



**RANCANG BANGUN SISTEM PEMANTAU SEPEDA ANAK  
BERBASIS IOT MENGGUNAKAN APLIKASI ANDROID**

**“PERANCANGAN SISTEM MIKROKONTROLER PEMANTAU  
SEPEDA ANAK”**

**TUGAS AKHIR**

**BERLIAN SATRIO SEJATI**

**1903332010**

**PROGRAM STUDI TEKNIK 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 PEMANTAU SEPEDA ANAK  
BERBASIS IOT MENGGUNAKAN APLIKASI ANDROID**

**“PERANCANGAN SISTEM MIKROKONTROLER PEMANTAU  
SEPEDA ANAK”**

**TUGAS AKHIR**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar  
Diploma Tiga**

**POLITEKNIK  
NEGERI  
JAKARTA**

**BERLIAN SATRIO SEJATI**

**1903332010**

**PROGRAM STUDI TEKNIK 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 benar.

Nama : Berlian Satrio Sejati

NIM : 1903332010

Tanda Tangan :



Tanggal : 3 Agustus 2022



POLITEKNIK  
NEGERI  
JAKARTA

© 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 mengumumkannya 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

LEMBAR PENGESAHAN  
TUGAS AKHIR

Tugas Akhir diajukan oleh :

Nama Mahasiswa : Berlian Satrio Sejati  
Nomor Induk Mahasiswa : 1903332010  
Program Studi : Telekomunikasi  
Judul Tugas Akhir : Rancang Bangun Sistem Pemantau Sepeda Anak  
Berbasis IoT Menggunakan Aplikasi Android

Telah diuji oleh tim penguji dalam Sidang Tugas Akhir pada 5 Agustus 2022 dan dinyatakan **LULUS**.


Pembimbing : Toto Supriyanto, S.T., M.T.  
NIP. 19660306 199003 1 001

()

Depok, 24 Agustus 2022

Disahkan oleh  
Ketua Jurusan Teknik Elektro



  
Ir. Sri Danaryani, M.T.  
NIP. 19630503 199103 2 001



## KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, penulis dapat menyelesaikan Tugas Akhir ini. Penulisan Tugas Akhir yang berjudul Rancang Bangun Sistem Pemantau Sepeda Anak Berbasis IoT Menggunakan Aplikasi Android dengan sub-judul “Perancangan Sistem Mikrokontroler Pemantau Sepeda Anak” Penulisan Tugas Akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Diploma Tiga Politeknik.

Penulis menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan tugas akhir ini, sangatlah sulit bagi penulis untuk menyelesaikan tugas akhir ini. Oleh karena itu, penulis mengucapkan terima kasih kepada :

1. Toto Supriyanto, S.T., M.T. selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan penulis dalam penyusunan tugas akhir ini;
2. Seluruh Staf Pengajar dan Karyawan Jurusan Teknik Elektro Politeknik Negeri Jakarta, khususnya Program Studi Telekomunikasi;
3. Ibu, Bapak, Abang dan Kakak Tingkat penulis yang telah memberikan bantuan dukungan material dan moral; dan
4. Hanunnisa Nur Rahmahyanti, selaku rekan Tugas Akhir serta rekan-rekan satu prodi Telekomunikasi angkatan 2019 yang telah saling mendukung dan bekerja sama demi menyelesaikan Tugas Akhir ini.

Akhir kata, penulis berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga Tugas Akhir ini membawa manfaat bagi pengembangan ilmu.

Depok, Agustus 2022

Penulis



## Rancang Bangun Sistem Pemantau Sepeda Anak Berbasis IoT Menggunakan Aplikasi Android

### “Perancangan Sistem Mikrokontroler Pemantau Sepeda Anak”

#### Abstrak

Bersepeda merupakan aktivitas berolahraga dan bermain anak bahkan sebagai moda transportasi untuk pergi ke sekolah, Taman Pendidikan al-Qur'an (TPQ), dan les atau belajar bersama. Saat bersepeda, hal-hal yang tidak diinginkan bisa saja terjadi pada anak seperti kecelakaan, tersesat, diculik, atau sepeda hilang dicuri. Salah satu faktor yang menyebabkan kecelakaan anak saat bersepeda adalah kerap kali anak bermain balap sepeda dengan teman-temannya atau bersepeda dengan kecepatan yang tidak wajar dalam usianya tanpa pengawasan orang tua. Orang tua tak selalu bisa mengawasi anak-anaknya yang bersepeda dikarenakan kesibukan pekerjaan dan keterbatasan jarak. Untuk mengatasi hal tersebut maka dibuat Sistem Pemantau Sepeda Anak Berbasis Internet of Things (IoT) Menggunakan Aplikasi Android. Sistem ini dapat memantau kondisi sepeda ketika sepeda terjatuh dan lokasi terjatuh maupun ketika melebihi kecepatan. Arduino Mega sebagai pusat kontrol data dari input yang berupa GPS NEO M8N, ESP32, Sensor HW-201, Sensor MPU6050, serta output yaitu Serial MP3 Player. Pembacaan GPS berdasarkan latitude dan longitude yang dikirimkan ke firebase, sensor MPU6050 membaca nilai  $\geq 55^\circ$  maka sepeda terindikasi terjatuh dalam rentang nilai  $56^\circ - 63^\circ$ . Sedangkan sensor HW 201 melebihi  $\geq 25$  kmph maka sepeda mengirimkan notifikasi sepeda melewati batas kecepatan dan dalam nilai rentang 8 kmph – 20 kmph sepeda tidak melewati batas kecepatan. Ketika kecepatan 25 kmph maka sepeda akan terindikasi melewati batas kecepatan dan serial mp3 player akan mengeluarkan suara melalui speaker.

**Kata kunci:** Arduino Mega, ESP32, IoT, Sepeda

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 mengumumikan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



## Design of an IoT-Based Children Bicycle Monitoring System Using an Android Application

### "Design of a Children's Bicycle Monitoring Microcontroller System"

#### Abstract

Cycling is an activity to exercise and play with children, even as a mode of transportation to go to school, Al-Qur'an Education Park (TPQ), and take lessons or study together. When cycling, unwanted things can happen to children such as accidents, getting lost, being kidnapped, or bicycles being stolen. One of the factors that cause child accidents while cycling is that children often play bicycle racing with their friends or cycle at an unnatural speed for their age without parental supervision. Parents cannot always supervise their children who are cycling due to busy work and limited distance. To overcome this, an Internet of Things (IoT)-Based Child Bicycle Monitoring System was created using the Android Application. This system can monitor the condition of the bicycle when the bicycle is dropped and the location of the fall or when it exceeds speed. Arduino Mega as a data control center from inputs in the form of GPS NEO M8N, ESP32, Sensor HW-201, Sensor MPU6050, and output, namely Serial MP3 Player. GPS readings are based on latitude and longitude sent to firebase, MPU6050 sensor reading the value  $55^\circ$  then the bike is indicated to have fallen in the value range  $56^\circ - 63^\circ$ . While the HW 201 sensor if it exceeds 25 kmph, the bicycle sends a notification that the bicycle has passed the speed limit and in the value range of 8 kmph - 20 kmph the bicycle does not exceed the speed limit. When the speed is 25 kmph, the bicycle will be indicated as crossing the speed limit and the serial mp3 player will make a sound through the speaker.

**Keywords:** Arduino Mega, Bicycle, ESP32, IoT,

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 mengumumikan 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 mengumumikan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR ISI

	Halaman
HALAMAN JUDUL .....	ii
HALAMAN PERNYATAAN ORISINALITAS .....	iii
LEMBAR PENGESAHAN .....	iv
KATA PENGANTAR.....	v
ABSTRAK .....	vi
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	x
DAFTAR TABEL .....	xi
DAFTAR LAMPIRAN .....	xii
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	2
1.3 Tujuan.....	2
1.4 Luaran .....	2
<b>BAB II TINJAUAN PUSTAKA</b> .....	<b>3</b>
2.1 <i>Internet of Things</i> (IoT) .....	3
2.2 Unsur-Unsur <i>Internet of Things</i> (IoT) .....	3
2.3 Cara Kerja <i>Internet of Things</i> (IoT).....	3
2.4 Arduino Mega 2560.....	4
2.5 ESP32 .....	5
2.6 Sensor MPU6050.....	5
2.7 Modul GPS NEO M8N.....	6
2.8 HW-201 (IR Proximity).....	6
2.9 Serial MP3 Player.....	7
2.10 <i>Battery Management System</i> (BMS) .....	8
2.11 Baterai 18650.....	8
2.12 Speaker .....	9
2.13 Sepeda.....	9
2.14 Arduino IDE .....	10
2.15 Firebase.....	13
2.16 GPS.....	13
2.17 <i>Latitude</i> dan <i>Longitude</i> .....	13
2.18 Google Maps.....	14
<b>BAB III PERANCANGAN DAN REALISASI</b> .....	<b>15</b>
3.1 Deskripsi Sistem .....	15
3.1.1 Cara Kerja Sistem.....	16
3.1.2 Spesifikasi Sistem .....	17
3.1.3 Diagram Blok .....	18
3.2 Realisasi Sistem.....	19





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

3.2.1	Realisasi Sistem Operasi Pemantau Sepeda Anak .....	19
3.2.1.1	Realisasi Sensor MPU 6050 .....	20
3.2.1.2	Realisasi GPS NEO M8N .....	20
3.2.1.3	Realisasi Serial MP3 Player .....	21
3.2.1.4	Realisasi HW-201 (IR Proximity).....	21
3.2.1.5	Realisasi ESP32.....	22
3.2.2	Pemrograman Arduino .....	23
3.2.2.1	Penambahan Library .....	24
3.2.2.2	Program Arduino IDE .....	24
3.2.2.3	Pemrograman ESP32 .....	32
3.2.3	Realisasi Catu Daya .....	36
<b>BAB IV</b>	<b>PEMBAHASAN.....</b>	<b>37</b>
4.1	Pengujian <i>Output</i> Catu Daya BMS.....	37
4.1.1	Deskripsi Pengujian.....	37
4.1.2	Prosedur Pengujian.....	37
4.1.3	Data Hasil Pengujian.....	38
4.1.4	Analisis Data .....	38
4.2	Pengujian Modul GPS NEO M8N.....	38
4.2.1	Deskripsi Pengujian.....	39
4.2.2	Prosedur Pengujian.....	39
4.2.3	Data Hasil Pengujian.....	39
4.2.4	Analisis Data .....	40
4.3	Pengujian Sensor MPU-6050 .....	40
4.3.1	Deskripsi Pengujian.....	40
4.3.2	Prosedur Pengujian.....	40
4.3.3	Data Hasil Pengujian.....	40
4.3.4	Analisis Data .....	41
4.4	Pengujian Sensor HW-201 IR Proximity , Serial <i>MP3 Player</i> dan <i>Speaker</i> .....	41
4.4.1	Deskripsi Pengujian.....	41
4.4.2	Prosedur Pengujian.....	41
4.4.3	Data Hasil Pengujian.....	42
4.4.4	Analisis Data .....	42
<b>BAB V</b>	<b>PENUTUP.....</b>	<b>43</b>
5.1	Simpulan.....	43
5.2	Saran .....	43
<b>DAFTAR PUSTAKA .....</b>	<b>44</b>	
<b>DAFTAR RIWAYAT HIDUP .....</b>	<b>46</b>	
<b>LAMPIRAN</b>		



## DAFTAR GAMBAR

	Halaman
Gambar 2.1 Cara Kerja Internet of Things .....	4
Gambar 2.2 Arduino Mega 2560 .....	4
Gambar 2.3 ESP32 .....	5
Gambar 2.4 Sensor MPU-6050 .....	5
Gambar 2.5 Modul GPS NEO M8N .....	6
Gambar 2.6 HW-201 .....	7
Gambar 2.7 DFPlayer.....	8
Gambar 2.8 BMS .....	8
Gambar 2.9 Baterai 18650 .....	9
Gambar 2.10 Speaker .....	9
Gambar 2.11 Sepeda .....	10
Gambar 2.12 Arduino IDE.....	11
Gambar 3.1a Modul dalam <i>Casing</i> .....	16
3.1b Ilustrasi Perancangan Pemantau Sepeda Anak Berbasis IoT.....	16
Gambar 3.2 Diagram Alir Sistem Pemantau Sepeda Anak Berbasis IoT.....	17
Gambar 3.3 Diagram Blok Sistem Pemantau Sepeda Anak Berbasis IoT.....	18
Gambar 3.4 Skematik Komponen Sistem Pemantau Sepeda Anak .....	19
Gambar 3.5 Realisasi Sensor MPU-6050 pada Arduino Mega .....	20
Gambar 3.6 Realisasi GPS NEO M8N pada Arduino Mega .....	21
Gambar 3.7 Realisasi Serial MP3 Player pada Arduino Mega .....	21
Gambar 3.8 Realisasi HW-201 (IR Proximity) pada Arduino Mega.....	22
Gambar 3.9 Realisasi ESP32 pada Arduino Mega.....	22
Gambar 3.10 Diagram Alir Pemrograman Arduino.....	23
Gambar 3.11 Skematik Catu Daya.....	36
Gambar 4.1 Hasil Pengukuran Tegangan VDC dari <i>Battery Shield</i> .....	38

### 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 mengumumikan 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

DAFTAR TABEL

	Halaman
Tabel 2.1 Rata-Rata Kecepatan Bersepeda berdasarkan Umur Pengendaranya ...	10
Tabel 3.1 Spesifikasi Sistem dan Alat .....	18
Tabel 3.2 Pin Komponen dengan Pin Arduino Mega .....	19
Tabel 4.1 Hasil Pengujian Modul GPS NEO M8N .....	39
Tabel 4.2 Hasil Pengujian Sensor MPU-6050 .....	41
Tabel 4.3 Pengujian Sensor <i>Infrared</i> dan <i>Serial MP3 Player</i> yang terhubung dengan <i>Speaker</i> .....	42





## © Hak Cipta milik Politeknik Negeri Jakarta

### DAFTAR LAMPIRAN

- L-1 Skematik Rangkaian Sistem
- L-2 Skematik Rangkaian Catu Daya
- L-3 Desain *Casing*
- L-4 Pemrograman Arduino Mega 2560
- L-5 Pemrograman ESP32
- L-6 *Datasheet* Arduino Mega 2560
- L-7 *Datasheet* ESP32
- L-8 *Datasheet* MPU-6050
- L-9 *Datasheet* HW-201
- L-10 *Datasheet Serial MP3 Player*
- L-11 *Datasheet* GPS NEO M8N
- L-12 Dokumentasi



#### 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 kritik 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

## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Sejak pandemi Covid-19 masyarakat kian menyadari pentingnya gaya hidup sehat. Selain dengan memilih makan-makanan sehat, olahraga menjadi cara untuk menjaga kebugaran tubuh. Salah satu olahraga yang menarik minat di tengah situasi seperti ini adalah bersepeda. Peminatnya meningkat drastis karena berbagai kemudahannya. Selain itu, sepeda juga banyak jenisnya. Mulai dari sepeda *road bike*, *mountain bike*, hingga sepeda anak. Selain untuk berolahraga, bersepeda juga menjadi aktivitas bermain sehari-hari anak bahkan sebagai moda transportasi untuk pergi ke sekolah, Taman Pendidikan al-Qur'an (TPQ), dan les atau belajar bersama. Saat bersepeda, hal-hal yang tidak diinginkan bisa saja terjadi pada anak seperti kecelakaan, tersasar, diculik, atau sepeda hilang dicuri. Dikutip dari [Balicycling.com](http://Balicycling.com), Pakar Kesehatan Anak, Dr Kate M. Cronan, dari Jaringan Rumah Sakit Anak Nemours, melaporkan bahwa lebih dari 300.000 anak muda Amerika setiap tahun berakhir di ruang gawat darurat karena kecelakaan sepeda. Beberapa cedera yang kecil, namun beberapa anak mengalami cedera kepala atau patah tulang.

Salah satu faktor yang menyebabkan kecelakaan anak saat bersepeda adalah kerap kali anak bermain balap sepeda dengan teman-temannya atau dengan kata lain bersepeda dengan kecepatan yang tidak wajar dalam usianya tanpa pengawasan orang tua. Hal inilah yang mendasari pengusul untuk membuat sebuah sistem pemantau sepeda anak dengan tujuan untuk mempermudah orang tua dalam mengawasi anak mereka saat bersepeda. Sistem yang akan dibuat menggunakan input berupa sensor IR *Proximity*, sensor MPU-6050, dan modul GPS NEO M8N yang dihubungkan dengan mikrokontroler Arduino Mega 2560 dan ESP32, *DFPlayer mini* dan *speaker* sebagai outputnya yang semua sistemnya terintegrasi dengan aplikasi Android. Aplikasi Android ini akan menampilkan berapa kecepatan dan lokasi anak saat bersepeda serta memberikan notifikasi apabila sepeda anak terjatuh. Selain itu, sistem akan memberikan peringatan apabila kecepatan bersepeda melebihi batas wajar melalui *speaker* yang terpasang pada sepeda.



## © 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 mengumumkkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

### 1.2 Perumusan Masalah

Berdasarkan latar belakang yang diuraikan di atas, maka permasalahan yang akan dibahas pada tugas akhir ini adalah sebagai berikut.

1. Bagaimana cara membuat prototype pemantau sepeda anak dan pemrograman arduino?
2. Bagaimana pengujian pembacaan GPS, sensor MPU-6050, dan *DFPlayer mini* pada sepeda anak

### 1.3 Tujuan

Tujuan yang ingin dicapai pada tugas akhir ini adalah sebagai berikut.

1. Membuat *prototype* pemantau sepeda anak dan pemrograman arduino
2. Menguji *output* tegangan BMS, pembacaan GPS, sensor MPU-6050, sensor IR Proximity dan *Serial MP3 Player* pada sepeda

### 1.4 Luaran

Adapun luaran dari Tugas Akhir “Rancang Bangun Sistem Pemantau Sepeda Anak Berbasis IoT Menggunakan Aplikasi Android” ini adalah :

1. Produk alat Tugas Akhir
2. Laporan Tugas Akhir
3. Jurnal Ilmiah
4. Poster

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

## BAB V PENUTUP

Pada bab ini menjelaskan mengenai kesimpulan dan saran yang dapat diambil dari pembuatan tugas akhir ini.

### 5.1 Simpulan

Berdasarkan perancangan dan hasil pengujian dari alat Tugas Akhir yang telah dibuat, maka dapat disimpulkan bahwa :

- 1) Sistem pemantau sepeda anak direalisasikan dengan memprogram Arduino Mega 2560 sebagai pusat operasi. Hasil yang ditampilkan serial monitor berisi data pemantauan kecepatan sepeda dengan satuan km/h, titik lokasi berupa *latitude* dan *longitude*, dan kemiringan sepeda dengan satuan derajat ( $^{\circ}$ ) serta serial *mp3 player* mengeluarkan suara melalui speaker.
- 2) GPS NEO M8N yang terkoneksi sinyal GPS berhasil mendeteksi keberadaan sepeda anak *latitude* dan *longitude* sudah sesuai dengan lokasi pada aplikasi *google maps*; Sensor MPU-6050 berhasil mendeteksi kemiringan sepeda anak yang dapat mengindikasikan sepeda terjatuh dengan rentang nilai kemiringan sebesar  $56^{\circ}$  -  $63^{\circ}$  yang didapatkan saat pengujian; *Battery Management System* (BMS) sebagai pemberi daya untuk menyalakan dan mematikan sistem dengan keluaran daya sebesar 3,83 V; serta pada pengujian Sensor Infrared HW-201 didapatkan hasil pembacaan kecepatan sepeda anak sebesar 8 km/h - 20 km/h dan serial *mp3 player* tidak memutar suara melalui *speaker*. Namun ketika didapatkan nilai kecepatan sepeda sebesar 25 km/h *speaker* mengeluarkan suara dari serial *mp3 player*.

### 5.2 Saran

Pada saran kali ini yaitu dalam pembuatan *casing* pada alat ini lebih rapi lagi dan dalam peletakan komponen sepeda lebih rapih dan tertata dalam mengatur *wiring*. Ke depannya mikrokontroler yang digunakan dapat mengirim dan menerima data lebih cepat serta sensor lebih akurat dalam pengambilan data.



## DAFTAR PUSTAKA

- Aisa, S. (2021). Aplikasi Pencarian Bengkel Aktif dengan Google Maps API Berbasis Web. *Journal of Computer and Information Technology*, 4(2), 61-69. <http://e-journal.unipma.ac.id/index.php/doubleclick/article/view/8001/3151>
- Astuti, N.F. (2018). Mengenal Fungsi Speaker, Lengkap dengan Jenis-Jenisnya. <https://www.merdeka.com/jabar/mengenal-fungsi-speaker-lengkap-dengan-jenis-jenisnya-klm.html>
- Bryan, Rafael. (2022). Average Cycling Speed by Age Chart. <https://bikerhype.com/average-cycling-speed-by-age-chart/>
- Dickson. (2020). Pengertian Latitude dan Longitude (Garis Lintang dan Garis Bujur). <https://ilmupengetahuanumum.com/pengertian-latitude-dan-longitude-garis-lintang-dan-garis-bujur/>
- Edwinanto & Hasanah, N. (2019). Kit Elektro Guna Meningkatkan Layanan Praktikum Mahasiswa. *Jurnal Rekayasa Teknologi Nusa Putra*, 7(1), 1-23. <https://jurnal.nusaputra.ac.id/rekayasa/paper/70>
- Faudin, A. (2018). Cara Membuat MP3 Player Menggunakan Arduino. <https://www.nyebarilmu.com/cara-membuat-mp3-player-menggunakan-arduino/>
- Firdaus, R. (2020). Rancang Bangun Sistem Navigasi Multirotor Berbasis Waypoint Dan Computer Vision. <https://elibrary.unikom.ac.id/id/eprint/2844/>
- Nurnawati, E.K. dkk. (2018). Pemanfaatan Real Time Database untuk Aplikasi Berbasis Lokasi. *Prosiding Seminar Nasional Aplikasi Sains & Teknologi (SNAST) 2018*, 49-60. <https://ejournal.akprind.ac.id/index.php/prosidingsnast/article/view/1493/1185>
- Nuryaman, A., Mulyana, E., & Mardiaty, R. (2018). Rancang Bangun Prototipe Alat Pengukur Kecepatan Kendaraan dengan Sensor Infra Merah. *Prosiding - Seminar Nasional Teknik Elektro UIN Sunan Gunung Djati Bandung*, 345-366. <https://senter.ee.uinsgd.ac.id/repositori/index.php/prosiding/article/view/senter2017p39>
- Razor, A. (2020). Software Arduino IDE: Cara Download, Instal, dan Fungsinya. <https://www.aldyrazor.com/2020/05/software-arduino-ide.html>
- Riadi, M. (2020). Sepeda (Sejarah, Bagian, Prinsip Kerja, Jenis-Jenis, dan Manfaat). <https://www.kajianpustaka.com/2020/08/sepeda.html>

### 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 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 mengumumikan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

- Rinaldi, A. (2020). *Internet of Things*.  
<https://www.smkn4tangsel.sch.id/read/7/artikel-internet-of-things>
- Sandi. (2018). Baterai 18650. <https://www.sandielektronik.com/2018/12/baterai-18650.html>
- Setiawan, R. dkk. (2021). Gesture Control Menggunakan IMU MPU 6050 Metode Kalman Filter Sebagai Kendali Quadcopter. <https://aau.e-journal.id>
- Siregar, N.T., Khair, U., & Budiman, A. (2021). Rancang Bangun Alat Sterilisasi Penyemprotan Disinfektan Otomatis untuk Barang Online Shop Berbasis Arduino. *Prosiding SNASTIKOM: Seminar Nasional Teknologi Informasi & Komunikasi, 1(1)*, 203-210.  
<https://prosiding.snastikom.com/index.php/SNASTIKOM2020/article/view/120>
- Sulistio. (2021). Mikrokontroler ESP32.  
<https://raharja.ac.id/2021/11/16/mikrokontroler-esp32-3/>
- Widjaya, S.P. (2020). GPS: Pengertian, Sejarah, Manfaat, Cara Kerja, Komponen, dan Jenis. <https://tekno.foresteract.com/gps/>
- Wahyudi, dkk. (2021). Rancang Bangun Media Pembelajaran *Battery Management System (BMS)* untuk SMK Kompetensi Keahlian Teknik Kendaraan Ringan. *KoPen : Konferensi Pendidikan Nasional, 3(2)*, 241-248.  
[https://ejurnal.mercubuana-yogya.ac.id/index.php/Prosiding\\_KoPeN/article/view/2821](https://ejurnal.mercubuana-yogya.ac.id/index.php/Prosiding_KoPeN/article/view/2821)

**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 mengumumikan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



## DAFTAR RIWAYAT HIDUP

### Berlian Satrio Sejati

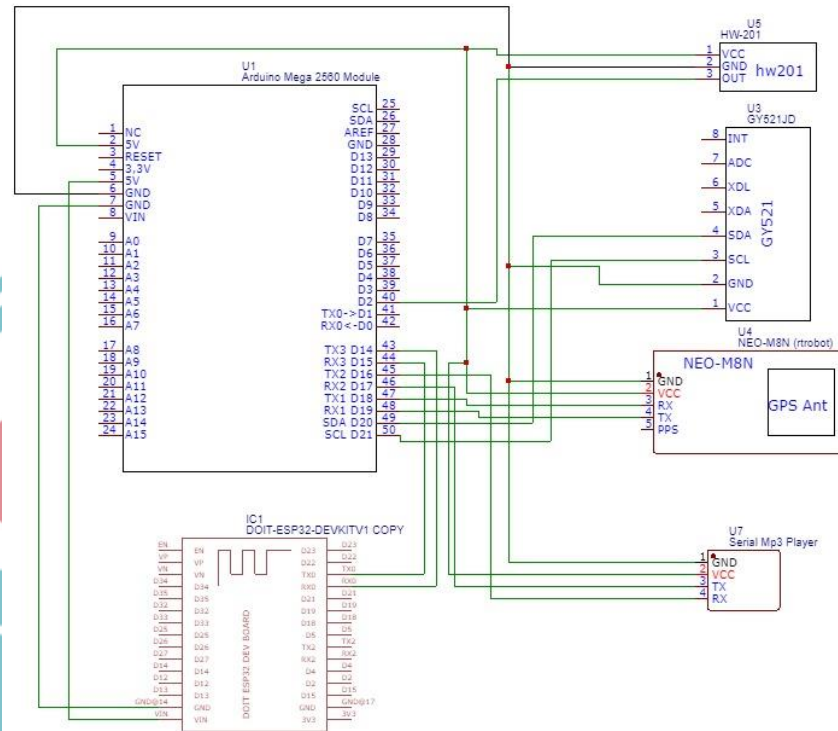
Lahir di Jakarta, 31 Agustus 2001. Lulus dari MIN 3 Cijantung tahun 2013, SMPN 103 Jakarta tahun 2016, dan SMAN 98 Jakarta tahun 2019. Gelar Diploma Tiga (D3) diperoleh pada tahun 2022 dari Program Studi Telekomunikasi, Jurusan Teknik Elektro, Politeknik Negeri Jakarta.





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
  - 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



01

# SKEMATIK RANGKAIAN SISTEM



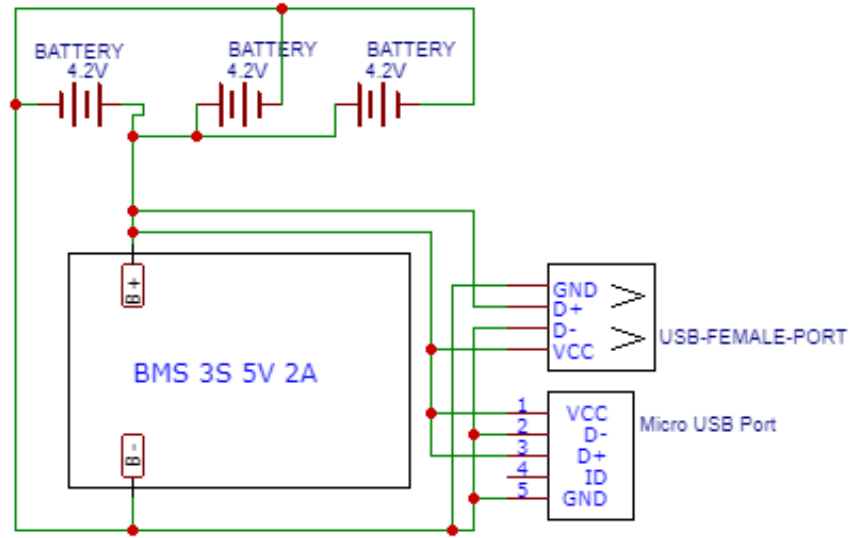
PROGRAM STUDI TELEKOMUNIKASI  
 JURUSAN TEKNIK ELEKTRO – POLITEKNIK NEGERI JAKARTA

Digambar	Berlian Satrio Sejati
Diperiksa	Toto Supriyanto, S.T., M.T.
Tanggal	Juli 2022



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  
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



02

## SKEMATIK RANGKAIAN CATU DAYA



PROGRAM STUDI TELEKOMUNIKASI  
JURUSAN TEKNIK ELEKTRO – POLITEKNIK NEGERI JAKARTA

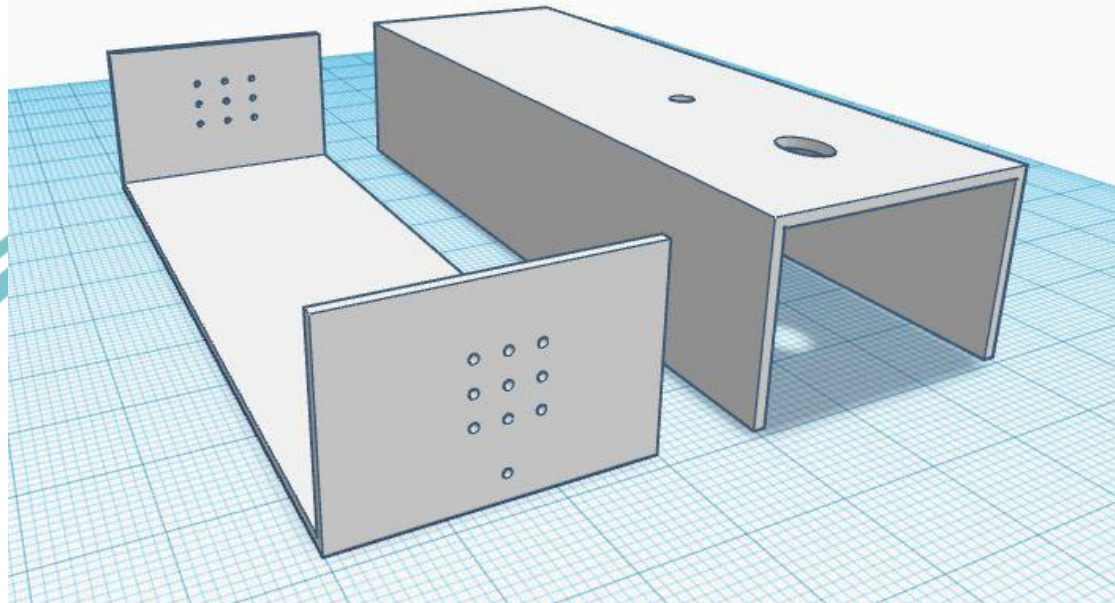
Digambar	Berlian Satrio Sejati
Diperiksa	Toto Supriyanto, S.T., M.T.
Tanggal	Juli 2022

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  
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



03

## DESAIN CASING



PROGRAM STUDI TELEKOMUNIKASI  
JURUSAN TEKNIK ELEKTRO – POLITEKNIK NEGERI JAKARTA

Digambar	Berlian Satrio Sejati
Diperiksa	Toto Supriyanto, S.T., M.T.
Tanggal	Juli 2022

POLITEKNIK  
NEGERI  
JAKARTA



```
#include <ArduinoJson.h>

#include "Wire.h"
#include <MPU6050_light.h>
MPU6050 mpu(Wire);
#define X_THRESHOLD_LEFT -55
#define X_THRESHOLD_RIGHT 55

#include <TinyGPSPlus.h>
static const uint32_t GPSTBaud = 9600;
TinyGPSPlus gps;
float gpslat = 0;
float gpslon = 0;
float gpsspeed = 0;
//String gpsgmapslink;
String str_gpslat;
String str_gpslon;

#include <TimerOne.h>
const int IRSensorPin = 2;
const int ledPin = 13;
int inputState;
int lastInputState = LOW;
long lastDebounceTime = 0;
long debounceDelay = 5;
long time;
long endTime;
long startTime;
int RPM = 0;
double trip = 0;
double kkbanspd = 0.00223;
float lnTime = 0;
float rBan = 0.311; //jari-jari ban sepeda dalam meter
float mpsSpeed = 0;
float kmphSpeed = 0;
int speedOverlimit = 0;

#include "SerialMP3Player.h"
#define GD3300_TX 16 //Arduino Mega TX Serial 2 pin
#define GD3300_RX 17 //Arduino Mega RX Serial 2 pin
SerialMP3Player mp3(GD3300_RX, GD3300_TX);

int fallState = 0;

// Generally, you should use "unsigned long" for variables that hold time
// The value will quickly become too large for an int to store
unsigned long mpuMillis = 0; // will store last time LED was
updated
unsigned long sendjsonMillis = 0;
// constants won't change:
const long mpuInterval = 10; // interval at which to blink
(milliseconds)
const long sendjsonInterval = 3000;

// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
  Serial3.begin(9600);
  Wire.begin();
  byte mpu_status = mpu.begin();
```

**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 mengumumikan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



## © 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 mengumumikan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
Serial.print(F("MPU6050 status: "));
Serial.println(mpu_status);
if (mpu_status != 0) {
    Serial.println("MPU6050 ERROR!!!");
}
Serial.println(F("Calculating offsets, do not move MPU6050"));
delay(1000);
mpu.calcOffsets(); // gyro and accelero
Serial.println("Done!\n");

Serial1.begin(GPSBaud);

Serial.println(F("DeviceExample.ino"));
Serial.println(F("A simple demonstration of TinyGPSPlus with an
attached GPS module"));
Serial.print(F("Testing TinyGPSPlus library v. "));
Serial.println(TinyGPSPlus::libraryVersion());
Serial.println(F("by Mikal Hart"));
Serial.println();

pinMode(IRSensorPin, INPUT);
pinMode(ledPin, OUTPUT);
delay(2000);
endTime = 0;
Timer1.initialize(1000000); // Set the timer to 60 rpm, 1,000,000
microseconds (1 second)
Timer1.attachInterrupt(timerIsr); // Attach the service routine here

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() {
    time = millis();

mpu.update();
    if ((millis() - mpuMillis) > mpuInterval) { // print data every 10ms
        Serial.print("X : ");
        Serial.print(mpu.getAngleX());
        Serial.print("\tY : ");
        Serial.print(mpu.getAngleY());
        Serial.print("\tZ : ");
        Serial.println(mpu.getAngleZ());
        if (mpu.getAngleX() < X_THRESHOLD_LEFT || mpu.getAngleX() >
X_THRESHOLD_RIGHT) {
            fallState = 1;
            sendJson();
            mpuInterval = 5000;
        } else {
            fallState = 0;
            mpuInterval = 10;
        }
        mpuMillis = millis();
    }

//GA PAKE PERINTAH DARI APK
// This sketch displays information every time a new sentence is
correctly encoded.
while (Serial1.available() > 0)
    if (gps.encode(Serial1.read()))
        displayInfo();
```



## © 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 mengumumikan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
if (millis() > 5000 && gps.charsProcessed() < 10)
{
    Serial.println(F("No GPS detected: check wiring."));
    while (true);
}

int currentSwitchState = digitalRead(IRSensorPin);

if (currentSwitchState != lastInputState) {
    lastDebounceTime = millis();
}

if ((millis() - lastDebounceTime) > debounceDelay) {
    if (currentSwitchState != inputState) {
        inputState = currentSwitchState;
        if (inputState == LOW) {
            digitalWrite(ledPin, LOW);
            calculateRPM(); // Real RPM from sensor
        }
        else {
            digitalWrite(ledPin, HIGH);
        }
    }
}
lastInputState = currentSwitchState;

if (millis() - sendJsonMillis >= sendJsonInterval) {
    sendJson();
}
}

//-----

void sendJson() {
    sendJsonMillis = millis();
    // Create the JSON document
    StaticJsonDocument<200> doc;
    doc["fallState"] = fallState;
    doc["lat"] = str_gpslat;
    doc["lon"] = str_gpslon;
    //doc["gmaps"] = gpsgmapslink;
    doc["speedometer"] = kmphSpeed;
    doc["speedOverlimit"] = speedOverlimit;

    // Send the JSON document over the "link" serial port
    serializeJson(doc, Serial3);
    serializeJson(doc, Serial);
    Serial.println();
}

//-----

void displayInfo()
{
    if (gps.location.isValid())
    {
        //    gpsslat = gps.location.lat();
        //    gpsslon = gps.location.lng();
        str_gpsslat = String(gps.location.lat(), 6);
        str_gpsslon = String(gps.location.lng(), 6);
        //gpsgmapslink = "www.google.com/maps/place/" + str_gpsslat + "," +
str_gpsslon;
        Serial.println(str_gpsslat);
        Serial.println(str_gpsslon);
        //    Serial.print(gps.location.lat(), 6);
```





## © 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

```
// Serial.print(F(", "));
// Serial.print(gps.location.lng(), 6);
}
}
//-----

void playSound() {
  delay(500);
  mp3.play(); // Play "hello.mp3".
  delay(6000); // Adjust the delay with the duration of the sound file
  playing
  mp3.stop();
  delay(500);
}
//-----

void calculateRPM() {
  startTime = lastDebounceTime;
  lnTime = startTime - endTime;
  RPM = 60000 / (startTime - endTime);
  mpsSpeed = rBan * ((2 * 3.14) / 60) * RPM;
  kmphSpeed = 3.6 * mpsSpeed;
  endTime = startTime;
  trip++;

  if (kmphSpeed >= 25.0) {
    speedOverlimit = 1;
    sendJson();
    playSound();
  } else if (kmphSpeed < 25.0) {
    speedOverlimit = 0;
    sendJson();
  }
}

void timerIsr()
{
  // Print RPM every second
  // RPM based on timer
  Serial.println("-----");
  time = millis() / 1000;
  Serial.println(time);
  Serial.print(" RPM: ");
  Serial.println(RPM);
  Serial.print(" mps: ");
  Serial.println(mpsSpeed);
  Serial.print(" kmph: ");
  Serial.println(kmphSpeed);
  Serial.print("Total Trip = ");
  Serial.println(trip);
  Serial.println("-----");

  delay(500);
  RPM = 0;
  mpsSpeed = 0;
  kmphSpeed = 0;
}
```



```
#include <ArduinoJson.h>

#if defined(ESP32)
#include <WiFi.h>
#elif defined(ESP8266)
#include <ESP8266WiFi.h>
#endif

#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>

/* 1. Define the WiFi credentials */
#define WIFI_SSID "POCO X3 Pro"
#define WIFI_PASSWORD "berlianss"

// For the following credentials, see
examples/Authentications/SignInAsUser/EmailPassword/EmailPassword.ino

/* 2. Define the API Key */
#define API_KEY "AIzaSyB4wqbnrLAjffZx8v28GaZl-BkzcYHkluKs"

/* 3. Define the RTDB URL */
#define DATABASE_URL "https://ta-berli-nunil-default-
rtbd.firebaseio.com/" //<databaseName>.firebaseio.com or
<databaseName>.<region>.firebasedatabase.app

/* 4. Define the user Email and password that already registered or added
in
your project */
#define USER_EMAIL "tatanu42@gmail.com"
#define USER_PASSWORD "tatanu100"

// Define Firebase Data object
FirebaseData fbdo;

FirebaseAuth auth;
FirebaseConfig config;

unsigned long sendDataPrevMillis = 0;

int count = 0;

// #include <NTPClient.h>
#include <WiFiUdp.h>

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

void setup()
{
  Serial.begin(9600);
  Serial.println();
  Serial.println();

  // pinMode(apkReqPin, OUTPUT);
```

**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 mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



## © 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

```
WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
Serial.print("Connecting to Wi-Fi");
while (WiFi.status() != WL_CONNECTED)
{
  Serial.print(".");
  delay(300);
}
Serial.println();
Serial.print("Connected with IP: ");
Serial.println(WiFi.localIP());
Serial.println();

Serial.printf("Firebase Client v%s\n\n", FIREBASE_CLIENT_VERSION);

/* Assign the api key (required) */
config.api_key = API_KEY;

/* Assign the user sign in credentials */
auth.user.email = USER_EMAIL;
auth.user.password = USER_PASSWORD;

/* Assign the RTDB URL (required) */
config.database_url = DATABASE_URL;

/* Assign the callback function for the long running token generation
task */
config.token_status_callback = tokenStatusCallback; // see
addons/TokenHelper.h

// Or use legacy authenticate method
// config.database_url = DATABASE_URL;
// config.signer.tokens.legacy_token = "<database secret>";

// To connect without auth in Test Mode, see
Authentications/TestMode/TestMode.ino

Firebase.begin(&config, &auth);

Firebase.reconnectWiFi(true);
#if defined(ESP8266)
fbdo.setBSSLBufferSize(512, 2048);
#endif
}

void ntp_update() {
  timeClient.update();

  epochTime = timeClient.getEpochTime();
  Serial.print("Epoch Time: ");
  Serial.println(epochTime);

  formattedTime = timeClient.getFormattedTime();
  Serial.print("Formatted Time: ");
  Serial.println(formattedTime);

  currentHour = timeClient.getHours();
  // Serial.print("Hour: ");
  // Serial.println(currentHour);

  currentMinute = timeClient.getMinutes();
  // Serial.print("Minutes: ");
  // Serial.println(currentMinute);

  currentSecond = timeClient.getSeconds();
  // Serial.print("Seconds: ");
```



## © 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 mengumumikan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
// Serial.println(currentSecond);

weekDay = weekDays[timeClient.getDay()];
// Serial.print("Week Day: ");
// Serial.println(weekDay);

//Get a time structure
struct tm *ptm = gmtime ((time_t *)&epochTime);

monthDay = ptm->tm_mday;
// Serial.print("Month day: ");
// Serial.println(monthDay);

currentMonth = ptm->tm_mon + 1;
// Serial.print("Month: ");
// Serial.println(currentMonth);

currentMonthName = months[currentMonth - 1];
// Serial.print("Month name: ");
// Serial.println(currentMonthName);

currentYear = ptm->tm_year + 1900;
// Serial.print("Year: ");
// Serial.println(currentYear);

//Print complete date:
currentDate = String(monthDay) + "-" + String(currentMonthName) + "-" +
String(currentYear);
Serial.print("Current date: ");
Serial.println(currentDate);

datetime = String(currentDate) + " " + String(formattedTime);
Serial.print("DATETIME: ");
Serial.println(datetime);

Serial.println("");
}

void loop()
{
  sendDataPrevMillis = millis();
*/
// 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<300> doc;

  // Read the JSON document from the "link" serial port
  DeserializationError err = deserializeJson(doc, Serial);

  if (err == DeserializationError::Ok)
  {
    int fallState = doc["fallState"];
    const char *gpslat = doc["lat"];
    const char *gpslon = doc["lon"];
    //const char *gmaps = doc["gmaps"];
    float speedometer = doc["speedometer"];
    int speedOverlimit = doc["speedOverlimit"];

    Serial.println(fallState);
    Serial.println(gpslat);
```



## © 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 mengumumikan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
Serial.println(gpslon);
//Serial.println(gmaps);
Serial.println(speedometer);
Serial.println(speedOverlimit);
Serial.println();

if (Firebase.ready()) {
  /*timeClient.update();

  time_t epochTime = timeClient.getEpochTime();
  Serial.print("Epoch Time: ");
  Serial.println(epochTime);
  */
  Serial.println("");
  Serial.print("Set int async... ");

  // Firebase.RTDB.setIntAsync(&fbdo, "/epoch", (unsigned
long)epochTime);
  Firebase.RTDB.setIntAsync(&fbdo, "/fallState", fallState);
  Firebase.RTDB.setStringAsync(&fbdo, "/lat", gpslat);
  Firebase.RTDB.setStringAsync(&fbdo, "/lon", gpslon);
  //Firebase.RTDB.setStringAsync(&fbdo, "/gmaps", gmaps);
  Firebase.RTDB.setFloatAsync(&fbdo, "/speedometer", speedometer);
  Firebase.RTDB.setIntAsync(&fbdo, "/speedOverlimit",
speedOverlimit);

  Serial.println("ok");
}
}
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();
}
}
```

**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

## Technical Specification

EAGLE files: [\\_arduino-mega2560-reference-design.zip](#) Schematic: [arduino-mega2560-schematic.pdf](#)

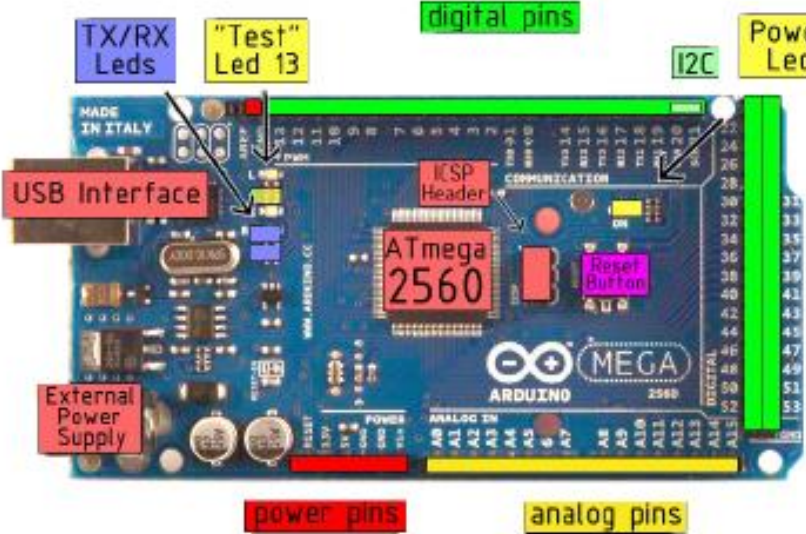
---





### Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

---

### the board





 radiospares  

**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

2 Pin Definitions

2 Pin Definitions

2.1 Pin Layout

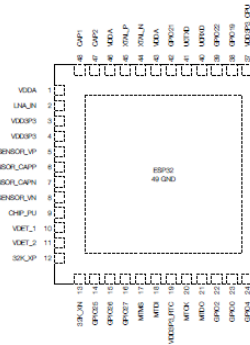


Figure 2: ESP32 Pin Layout (QFN 6x6, Top View)

3 Functional Description

3 Functional Description

This chapter describes the functions integrated in ESP32.

3.1 CPU and Memory

3.1.1 CPU

ESP32 contains one or two low-power Xtensa® 32-bit LX6 microprocessor(s) with the following features:

- 7-stage pipeline to support the clock frequency of up to 240 MHz (160 MHz for ESP32-S0WD)
- 16/24-bit Instruction Set provides high code-density
- Support for Floating Point Unit
- Support for DSP instructions, such as a 32-bit multiplier, a 32-bit divider, and a 40-bit MAC
- Support for 32 interrupt vectors from about 70 interrupt sources

The single-/dual CPU interfaces include:

- Xtensa RAM/ROM interface for instructions and data
- Xtensa Local Memory Interface for fast peripheral register access
- External and internal interrupt sources
- JTAG for debugging

3.1.2 Internal Memory

ESP32's internal memory includes:

- 448 KB of ROM for booting and core functions
- 620 KB of on-chip SRAM for data and instructions
- 8 KB of SRAM in RTC, which is called RTC FAST Memory and can be used for data storage; it is accessed by the main CPU during RTC Boot from the Deep-sleep mode.
- 8 KB of SRAM in RTC, which is called RTC SLOW Memory and can be accessed by the ULP coprocessor during the Deep-sleep mode.
- 1 Kbit of eFuse: 256 bits are used for the system (MAC address and chip configuration) and the remaining 768 bits are reserved for customer applications, including flash-encryption and chip-ID.
- Embedded flash or PSRAM

**Note:** Products in the ESP32 series differ from each other, in terms of their support for embedded flash or PSRAM and the size of them. For details, please refer to Section 7 for Number and Ordering Information.

5 Electrical Characteristics

5 Electrical Characteristics

5.1 Absolute Maximum Ratings

Stresses beyond the absolute maximum ratings listed in the table below may cause permanent damage to the device. These are stress ratings only, and do not refer to the functional operation of the device that should follow the recommended operating conditions.

Table 13: Absolute Maximum Ratings

Symbol	Parameter	Min	Max	Unit
VDDA, VDDSP3, VDDSP3_RTC, VDDSP3_CPU, VDD_SDIO	Voltage applied to power supply pins per power domain	-0.3	3.6	V
I <sub>output</sub>	Cumulative IO output current	-	1200	mA
T <sub>storage</sub>	Storage temperature	-40	150	°C

\* The chip worked properly after a 24-hour test in ambient temperature at 25 °C, and the I/Os in these domains (VDDSP3\_RTC, VDDSP3\_CPU, VDD\_SDIO) output high logic level to ground.

5.2 Recommended Operating Conditions

Table 14: Recommended Operating Conditions

Symbol	Parameter	Min	Typ	Max	Unit
VDDA, VDDSP3_RTC, VDDSP3_CPU, VDD_SDIO (3.3 V mode)	Voltage applied to power supply pins per power domain	2.3 <sub>(0.4)</sub>	3.3	3.6	V
VDDSP3_CPU	Voltage applied to power supply pin	1.8	3.3	3.6	V
I <sub>vdd</sub>	Current delivered by external power supply	0.5	-	-	A
T <sub>min</sub>	Operating temperature	-40	-	125	°C

1. When writing eFuse, VDDSP3\_RTC should be at least 3.3 V.
2. VDD\_SDIO works as the power supply for the related IO, and also for an external device. Please refer to the Appendix IO\_MUX of this datasheet for more details.
  - VDD\_SDIO can be sourced internally by the ESP32 from the VDDSP3\_RTC power domain:
    - When VDD\_SDIO operates at 3.3 V, it is often directly by VDDSP3\_RTC through a 6 Ω resistor, therefore, there will be some voltage drop from VDDSP3\_RTC.
    - When VDD\_SDIO operates at 1.8 V, it can be generated from ESP32's internal LDO. The maximum current the LDO can offer is 40 mA, and the output voltage range is 1.05 V~2.0 V.
  - VDD\_SDIO can also be driven by an external power supply.
3. Please refer to Power Scheme, section 2.3, for more information.
  - Chips with a 3.3 V flash or PSRAM (unpacked): the minimum voltage is 3.0 V.
  - Chips with no flash or PSRAM (unpacked): the minimum voltage is 2.3 V.
4. For more information, see Table 25 (ESP32 Ordering Information).
  - The operating temperature of ESP32-L1W0CH ranges from -40 °C to 105 °C, due to the flash embedded in it.
  - The operating temperature of ESP32-G0W0R0-V0 ranges from -40 °C to 85 °C, due to the PSRAM embedded in it.
  - The other chips in this series have no embedded flash or PSRAM, so their range of operating temperatures is -40 °C~105 °C.



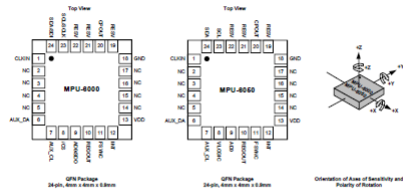
- Hak Cipta :**
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
    - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
    - b. Pengutipan tidak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

L-8 Datasheet MPU6050

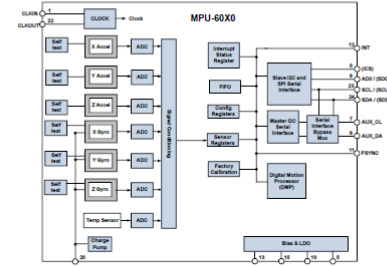
7 Applications Information

7.1 Pin Out and Signal Description

Pin Number	MPU-6000	MPU-6050	Pin Name	Pin Description
1	Y	Y	CLKIN	Optional external reference clock input. Connect to GND if unused.
6	Y	Y	AUX_DA	FC master serial data for connecting to external sensors
7	Y	Y	AUX_CL	FC master serial clock for connecting to external sensors
8	Y	Y	CS	SPH chip select (push-pull mode)
9	Y	Y	VDDIO	Digital I/O supply voltage
10	Y	Y	AD0 / SDO	FC Slave Address LSR (ADC), SPH serial data output (SDO)
11	Y	Y	AD1	FC Slave Address LSR (ADC)
12	Y	Y	FBUSOUT	Regulator filter capacitor connection
13	Y	Y	FBVNC	Frame synchronization digital input. Connect to GND if unused.
14	Y	Y	INT	Interrupt digital output (drain pole or open-drain)
15	Y	Y	VDD	Power supply voltage and Digital I/O supply voltage
18	Y	Y	GND	Power supply ground
19, 21	Y	Y	REBOOT	Reserved. Do not connect.
20	Y	Y	CSOUT	Change pending capacitor connection
22	Y	Y	REBOOT	Reserved. Do not connect.
23	Y	Y	SCL / SDA1	IC2 serial data (SCL), SPH master clock (SDA1)
24	Y	Y	SCL	IC2 serial clock (SCL)
25	Y	Y	SDA / SDO	IC2 serial data (SDA), SPH serial data input (SDO)
26	Y	Y	SDA	IC2 serial data (SDA)
2, 3, 4, 5, 14, 15, 16, 17	Y	Y	NC	Not internally connected. May be used for PCB trace routing.



7.5 Block Diagram



7.6 Overview

- The MPU-60X0 is comprised of the following key blocks and functions:
- Three-axis MEMS rate gyroscope sensor with 16-bit ADCs and signal conditioning
  - Three-axis MEMS accelerometer sensor with 16-bit ADCs and signal conditioning
  - Digital Motion Processor (DMP) engine
  - Primary FC and SPI (MPU-6000 only) serial communications interfaces
  - Auxiliary FC serial interface for 3<sup>rd</sup> party magnetometer & other sensors
  - Clocking
  - Sensor Data Registers
  - FIFO
  - Interrupts
  - Digital-Output Temperature Sensor
  - Gyroscope & Accelerometer Self-Test
  - Bias and LDO
  - Charge Pump

6 Electrical Characteristics

6.1 Gyroscope Specifications

VDD = 2.375V-3.45V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T<sub>a</sub> = 25°C

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
GYROSCOPE SENSITIVITY	Full-Scale Range	±160	±200	±250	°/s	
	FS <sub>1</sub> (SEL1)	±1000	±1300	±1600	°/s	
	FS <sub>2</sub> (SEL2)	±2000	±2600	±3200	°/s	
	FS <sub>3</sub> (SEL3)	±4000	±5200	±6400	°/s	
Gyroscope ACC Word Length		16	16	16	bits	
Sensitivity Scale Factor	FS <sub>1</sub> (SEL1)	331	331	331	LSB/°/s	
	FS <sub>2</sub> (SEL2)	66.2	66.2	66.2	LSB/°/s	
	FS <sub>3</sub> (SEL3)	16.6	16.6	16.6	LSB/°/s	
	25°C	-3	0.2	+3	%	
Sensitivity Scale Factor Tolerance			±2	%		
Sensitivity Scale Factor Variation Over Temperature			±2	%		
Nonlinearity	Best fit straight line 25°C		±0.2		%	
Offset Accuracy			±0.2		%	
GYROSCOPE ZERO-RATE OUTPUT (ZRO)	Initial ZRO Tolerance	25°C	±0.0	±0.0	°/s	
	ZRO Variation Over Temperature	40°C to +85°C		±0.0	°/s	
	Power-Supply Sensitivity (1-10Hz)	Sine wave, 100mVpp, VDD±2.0V		±0.2	°/s	
	Power-Supply Sensitivity (10-20Hz)	Sine wave, 100mVpp, VDD±2.0V		±0.2	°/s	
	Power-Supply Sensitivity (20Hz - 100Hz)	Sine wave, 100mVpp, VDD±2.0V		±0.4	°/s	
	Linear Acceleration Sensitivity	Static		±0.1	°/s	°/s/g
SELF-TEST RESPONSE	Relative	Change from factory trim	-14	14	%	1
	Offset				°/s	
GYROSCOPE NOISE PERFORMANCE	FS <sub>1</sub> (SEL1)	CLFPCG±2 (100Hz)	0.05	0.05	°/s-√Hz	
	Total RMS Noise	Bandwidth: 1Hz to 10Hz at 10Hz	0.033	0.033	°/s-√Hz	
	Low-Frequency RMS Noise		0.033	0.033	°/s-√Hz	
	Rate Noise Spectral Density		0.033	0.033	°/s-√Hz	
GYROSCOPE MECHANICAL FREQUENCIES	X-Axis		30	33	kHz	
	Y-Axis		27	30	kHz	
	Z-Axis		24	27	kHz	
LOW PASS FILTER RESPONSE	Programmable Range	5	256	Hz		
OUTPUT DATA RATE	Programmable	4	8,000	Hz		
GYROSCOPE START-UP TIME	DLFPCG±3		30	ms		
ZRO Settling Time (power-on)			30	ms		

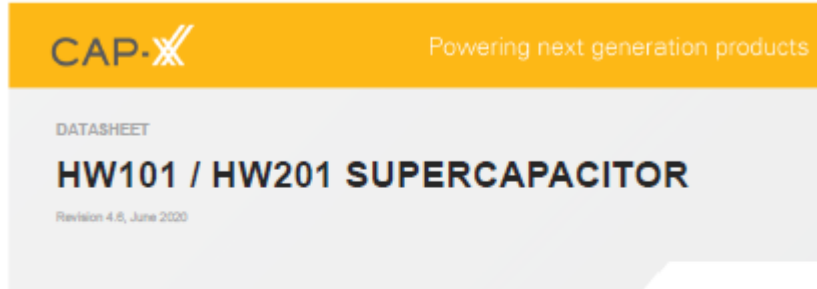
6.2 Accelerometer Specifications

VDD = 2.375V-3.45V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T<sub>a</sub> = 25°C

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
ACCELEROMETER SENSITIVITY	Full-Scale Range	±16g	±20g	±25g	g	
	FS <sub>1</sub> (SEL1)	±1000	±1300	±1600	g	
	FS <sub>2</sub> (SEL2)	±2000	±2600	±3200	g	
	FS <sub>3</sub> (SEL3)	±4000	±5200	±6400	g	
ADC Word Length	Output in two's complement format	16	16	16	bits	
Sensitivity Scale Factor	FS <sub>1</sub> (SEL1)	16,384	16,384	16,384	LSB/g	
	FS <sub>2</sub> (SEL2)	8,192	8,192	8,192	LSB/g	
	FS <sub>3</sub> (SEL3)	4,096	4,096	4,096	LSB/g	
	25°C		±0.2		%	
Initial Calibration Tolerance			±0.2	%		
Sensitivity Change vs. Temperature	FS <sub>1</sub> (SEL1), 40°C to +85°C		±0.2	%/°C		
Nonlinearity	Best Fit Straight Line		±0.5	%		
Cross-Axis Sensitivity			±0.2	%		
ZERO-G OUTPUT	Initial Calibration Tolerance	X and Y axis	±0.0	±0.0	g	1
	Z axis		±0.0	±0.0	g	
	Zero-G Level Change vs. Temperature	X and Y axis, 0°C to +70°C		±0.5	g/°C	
Z axis, 0°C to +70°C			±0.5	g/°C		
SELF-TEST RESPONSE	Relative	Change from factory trim	-14	14	%	2
	Offset				g	
NOISE PERFORMANCE	Power Spectral Density		0.001	0.001	g-√Hz	
	Low Pass Filter Response	Programmable Range	5	260	Hz	
OUTPUT DATA RATE	Programmable Range	4	1,000	Hz		
INFLUENCE FUNCTION INCREMENT			32	mg/LSB		

1. Typical zero-g initial calibration tolerance value after MLL3 preconditioning
2. Please refer to the following document for further information on Self-Test: MPU-6000/MPU-6050 Register Map and Descriptions





Powering next generation products

DATASHEET

## HW101 / HW201 SUPERCAPACITOR

Revision 4.6, June 2020

### Electrical Specifications

The HW101 is a single cell supercapacitor. The HW201 is a dual cell supercapacitor with two HW101 cells in series, so HW201 capacitance = Capacitance of HW101/2 and HS230 ESR = 2 x HW101 ESR.

Table 1: Absolute Maximum Ratings

Parameter	Name	Conditions	Min	Typical	Max	Units
Terminal Voltage	V <sub>peak</sub>	HW101	0		2.9	V
		HW201			5.8	
Temperature <sup>1</sup>	T <sub>max</sub>		-40		+85	°C

Table 2: Electrical Characteristics

Parameter	Name	Conditions	Min	Typical	Max	Units
Terminal Voltage	V <sub>n</sub>	HW101	0		2.75	V
		HW201	0		5.5	
Capacitance	C	HW101	608	760	912	mF
		HW201	304	380	456	
ESR	ESR	HW101		50	60	mΩ
		HW201		100	120	
Leakage Current	I <sub>L</sub>	2.75V, 23°C 120hrs		1	2	μA
RMS Current	I <sub>RMS</sub>	23°C			3	A
Peak Current <sup>2</sup>	I <sub>p</sub>	23°C			30	A

<sup>1</sup>Max continuous operating temp = +70°C but can withstand excursions to +85°C.

<sup>2</sup>Non-repetitive current, single pulse to discharge fully charged supercapacitor.

Table 3: Thickness

HW101F	1.3mm	No adhesive tape on underside of the supercapacitor	HW101G	1.4mm	Adhesive tape on underside, release tape removed
HW201F	2.7mm		HW201G	2.8mm	

This datasheet should be read in conjunction with the [CAP-XX Supercapacitor Product Guide](#) which contains information common to our product lines.

- 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

L-10 Datasheet GPS NEO M9N

NEO-MBN - Hardware Integration Manual

## 2 Design

### 2.1 Pin description

Function	Pin	No.	I/O	Description	Remarks
Power	VCC	23		Supply Voltage	Provide clean and stable supply.
	GND	10,12,13,14		Ground	Assure a good GND connection to all GND pins of the module, preferably with a large ground plane.
	V_BCKP	22		Backup Supply Voltage	It is recommended to connect a backup supply voltage to V_BCKP in order to enable warm and hot start features on the positioning module. Otherwise, connect to VCC.
Antenna	VDD_USB	7		USB Power Supply	To use the USB interface, connect this pin to 3.0-3.6V. If no USB serial port used connect to GND.
	RF_IN	11	I	GNSS signal input from antenna	The connection to the antenna has to be equal to the PCB. Use a controlled impedance of 50 Ω to connect RF_IN to the antenna or antenna connector.
UART	VCC_RF	9	O	Output Voltage RF section	VCC_RF can be used to power an external active antenna.
	TxD	20	O	Serial Port TX MOSI	Communication interface. Can be programmed to TX Ready for DDC interface.
RxD	RxD	21	I	Serial Port / SPI MISO	Serial input. Internal pull-up resistor to VCC. Leave open if not used.
	USB_DM	5	I/O	USB D+ line	MISO. If pin 2 low => SPI MISO.
USB	USB_DP	6	I/O	USB D- line	Serial input. Internal pull-up resistor to VCC. Leave open if not used.
	TEMP_SILE	3	O	Temperature Signal	USB bidirectional communication pin. Leave open if unused.
System	UARTBOOT_M	1	I	Reserved	Configurable temperature signal (one pulse per second by default). Leave open if not used.
	EXTINT	4	I	External Interrupt	For factory service, updates and reconfiguration, leave OPEN. Internal interrupt pin. Internal pull-up resistor to VCC. Leave open if not used. Function is disabled by default.
RESERVED	15	-	-	Reserved	Leave open.
SDA	SDA	18	I/O	DDC Data / SPI CLK	DDC Data. If pin 2 low => SPI chip select.
	SCL	19	I/O	DDC Clock / SPI CS	DDC Clock. If pin 2 low => SPI clock.
ANT_ON	14	O	ANT_ON	Antenna control can be used to turn on and off an optional external GPS.	
RESET_N	8	I	Reset input	Reset input	
D_SIL	2	I	Switch the interface	Allows selecting UART/DC/USB interface	
RESERVED	16,17	-	-	Reserved	Leave open.

Table 2: NEO-MBN Pinout

USA-75029905 - R01      Advance Information      Design Page 11 of 28

NEO-MBN - Hardware Integration Manual

## 2.2 Minimal design

This is a minimal design for a NEO-MBN GNSS receiver.

Figure 4: NEO-MBN passive antenna design

## 2.3 Layout: Footprint and paste mask

Figure 3 describes the footprint and provides recommendations for the paste mask for NEO-MBN LCC modules. Those are recommendations only and not specifications. Note that the copper and solder masks have the same size and position.

To improve the wetting of the half die, reduce the amount of solder paste under the module and increase the volume outside of the module by defining the dimensions of the paste mask to form a T-shape (or equivalent) extending beyond the copper mask. For the stencil thickness, see section 4.2.

Consider the paste mask outline when defining the minimal distance to the next component. The exact geometry, distance, stencil thickness and solder paste volumes must be adapted to the specific production processes (e.g. soldering) of the customer.

Figure 5: NEO-MBN footprint

USA-75029905 - R01      Advance Information      Design Page 12 of 28

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

L-11 Datasheet Serial MP3 Player

CATALEX 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--YX5100. 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 (KHz): 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 LINO / Leonardo / Mega2560 / DUE

2 / 10

CATALEX 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); Mirco SDHC card(<=32G)			/
File system format	Fat16 / Fat32			/
Uart baud rate	9600			bps
Dimensions	49X24X8.5			mm
Net Weight	5			g

3 / 10

CATALEX 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.

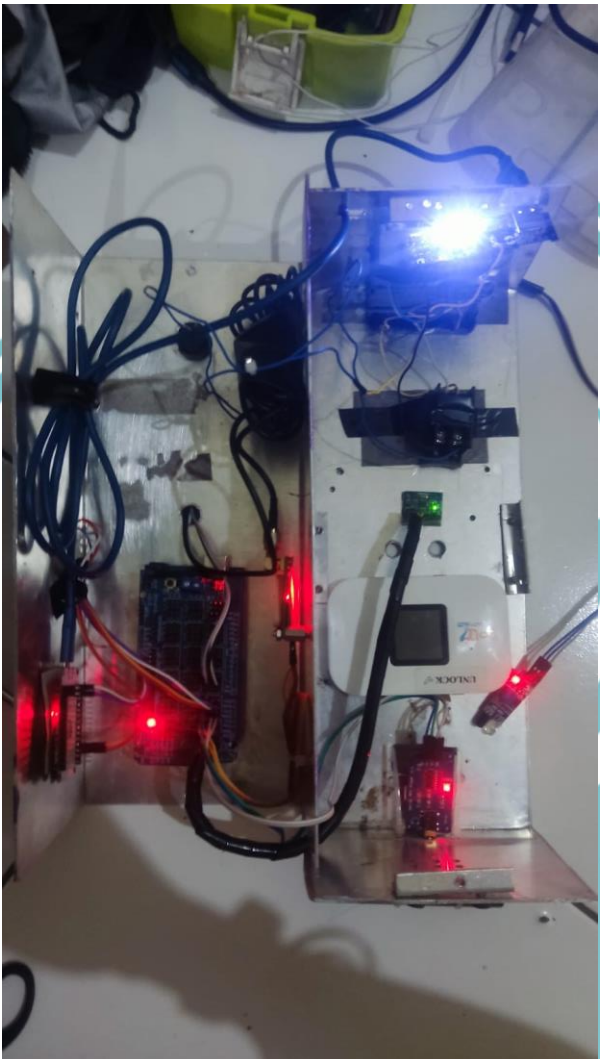
**Playbck indicator:** Green light. If it is ready to play or it is paused, it keeps lighting. If playing, it blinks.

**Headphone jacks:** 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





JAKARTA

K

## © 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 mengumumkkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

