



**MODUL PELATIHAN
MIKROKONTROLER I
EMBEDDED AND CYBER-PHYSICAL SYSTEM
LABORATORY**



Daftar Isi

Daftar Isi	i
Daftar Gambar	ii
BAB I Introduction to Microcontroller	1
1.1. Pengenalan	1
1.2. Macam-macam Mikrokontroler	3
1.3. Arsitektur Microcontroller	6
1.4. Spesifikasi Mikrokontroler	6
BAB II Konsep Dasar Sinyal Digital and Analog	10
2.1. Sinyal Digital vs Analog	10
2.2. Bilangan Digital	10
2.3. Konversi Bilangan Digital	12
2.4. Tegangan Logic Level	14
BAB III Fitur-Fitur Mikrokontroler	16
3.1. Analog to Digital Converter	16
3.2. Digital to Analog Converter	16
3.3. Pulse Width Modulation (PWM)	16
3.4. Timing & Clock	17
BAB IV Serial Komunikasi Mikrokontroler	19
4.1. Komponen Sistem Komunikasi	19
4.2. Universal Asynchronous Receiver/Transmitter	19
4.3. Inter Integrated Circuit	21
4.4. Serial Peripheral Interface	23
4.5. Wireless	25
BAB V Elektronika Daya pada Mikrokontroler	27
5.1. Pengenalan	27
5.2. Pengaplikasian Sumber tegangan DC	27
5.3. Pencegahan over-current pada mikrokontroler dan sensor	28
5.4. Filtrasi Noise dan Proteksi Tegangan untuk Sensor	29
5.5. Kesimpulan: Dos and Don'ts	30

Daftar Gambar

Gambar 1. Unit Mikrokontroler.....	1
Gambar 2. Mikroprosesor dan Mikrokontroler.....	1
Gambar 3. Perbedaan Struktur Mikroprosesor dan Mikrokontroler.....	2
Gambar 4. Berbagai Jenis Board Mikrokontroler Arduino	3
Gambar 5. Berbagai Macam Modul ESP8266.....	4
Gambar 6. Board NodeMCU ESP8266	4
Gambar 7. Board ESP32.....	5
Gambar 8. STM32F103C8T6 “Blue Pill”	5
Gambar 9. Diagram Arsitektur Mikrokontroler AVR	6
Gambar 10. Diagram Blok pada Datasheet Mikrokontroler.....	7
Gambar 11. Diagram Pinout pada Datasheet Mikrokontroler	8
Gambar 12. Tabel Konsumsi Energi pada Datasheet	8
Gambar 13. Pinout Mikrokontroler	9
Gambar 14. Pinout Modul Mikrokontroler Arduino Nano 33 IoT	9
Gambar 15. <i>Logic Level Threshold</i>	15
Gambar 16. Analog to Digital Converter dan Digital to Analog Converter.....	16
Gambar 17. <i>Pulse Width Modulation</i>	17
Gambar 18. Clock pada Mikrokontroler.....	18
Gambar 19. <i>Frequency Planning</i> pada sistem Mikrokontroler	18
Gambar 20. Skema Komunikasi UART	20
Gambar 21. Format Paket Data pada Komunikasi UART.....	20
Gambar 22. Diagram Waktu Protokol Komunikasi I2C.....	21
Gambar 23. Pemilihan Address pada protokol I2C	22
Gambar 24. Protokol I2C untuk Menulis Data Register.....	22
Gambar 25. Protokol I2C untuk Membaca Data Register	22
Gambar 26. Skema rangkaian I2C Arduino dengan sensor BMP180 dan MPU6050	23
Gambar 27. Diagram Komunikasi SPI antara Master dan Slave.....	24

Gambar 28. Konfigurasi Clock Polarity (CPOL) dan Clock Phase (CPHA)	24
Gambar 29. Elektronika Daya pada Mikrokontroler dan Sensor	24
Gambar 30. Macam-Macam Voltage Regulator	24
Gambar 31. Contoh Variable Switching Regulator	24
Gambar 32. Contoh Skematik Current Limiting Circuit	24
Gambar 33. Fuse dan Pengaplikasian di Rangkaian	24
Gambar 34. Contoh Pengaplikasian Filter pada Skematik	30
Gambar 35. Contoh Datasheet Electrical Characteristic Sensor	31
Gambar 36. Pengukuran Tegangan menggunakan Multimeter	32



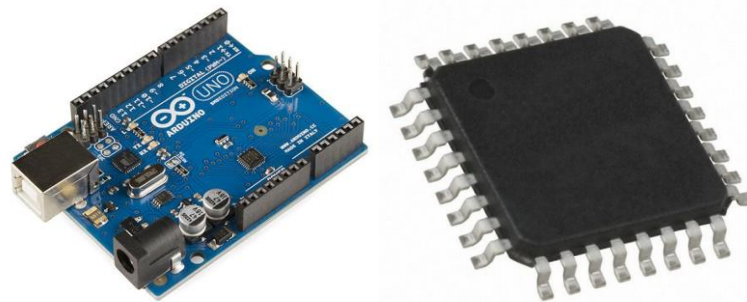
BAB I

Introduction to Microcontroller

1.1. Pengenalan

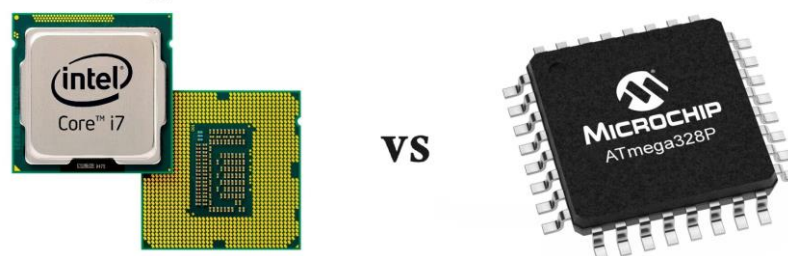
Unit mikrokontroler (MCU) pada dasarnya adalah komputer kecil dalam satu chip. Dirancang untuk mengelola tugas-tugas tertentu dalam sistem tertanam tanpa memerlukan sistem operasi yang kompleks. Mikrokontroler merupakan sebuah sistem mikroprosesor lengkap yang telah diintegrasikan menjadi satu kesatuan yang dapat diprogram dan dioperasikan secara mudah.

Mengintegrasikan pemrosesan, memori, dan peripheral input/output (I/O) secara bawaan. Mikrokontroler (MCU) cocok untuk aplikasi yang memerlukan pemrosesan sinyal waktu nyata serta berinteraksi dengan berbagai jenis sensor dan komunikasi antar perangkat.



Gambar 1. Unit Mikrokontroler

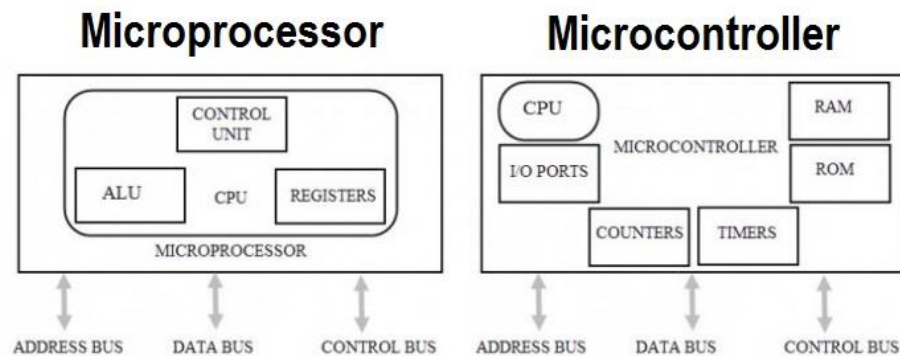
Microprocessor Microcontroller



Gambar 2. Mikroprosesor dan Mikrokontroler

Karakteristik utama dari sebuah mikrokontroler (MCU) adalah kombinasi dari semua elemen komputasi yang diperlukan ke dalam satu chip—mikrokontroler tidak memerlukan sirkuit eksternal tambahan untuk beroperasi. Sebaliknya, mikroprosesor

terdiri dari CPU dan beberapa chip pendukung yang menyediakan memori, antarmuka serial, peripheral input/output (I/O), dan fitur-fitur penting lainnya.



Microprocessor vs Microcontroller by EEEPROJECT.COM

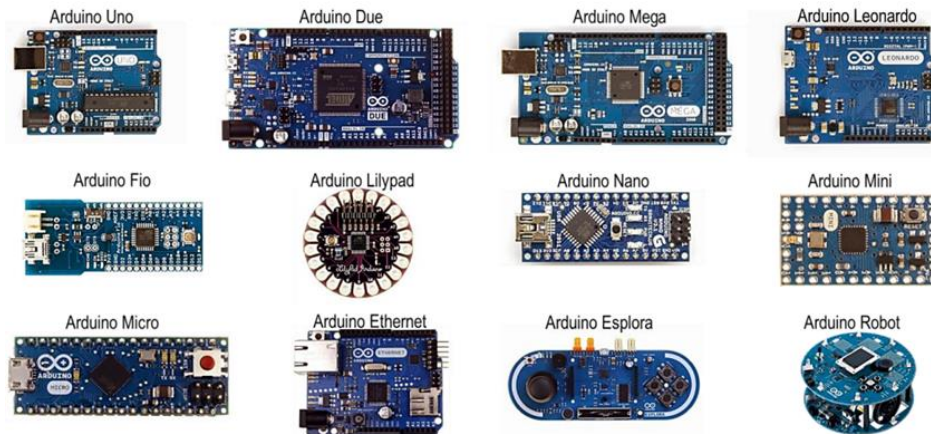
Gambar 3. Perbedaan Struktur Mikroprosesor dan Mikrokontroler

Sering dijumpai istilah mikroprosesor dan CPU digunakan secara bergantian, lebih akurat untuk menggambarkan mikroprosesor sebagai integrated circuit (IC) tunggal yang mengandung CPU dan dapat dihubungkan ke perangkat tambahan eksternal lainnya, seperti perangkat input/output. Perbedaan utama antara mikrokontroler dan mikroprosesor adalah bahwa mikrokontroler bersifat mandiri, sedangkan mikroprosesor dirancang untuk berinteraksi dengan perangkat tambahan eksternal lainnya.

Akibatnya, kegunaan dan fungsionalitas dari kedua perangkat ini juga berbeda bergantung pada situasi pengaplikasiannya. Pada tugas-tugas yang menuntut perangkat keras khusus dengan daya pemrosesan yang tinggi lebih cocok digunakan mikroprosesor. Sedangkan, pada tugas-tugas spesifik dalam sistem tertanam, seperti kontrol sensor atau motor, adalah contoh yang baik dari aplikasi mikrokontroler (MCU).

1.2. Macam-macam Mikrokontroler

1. Arduino



Gambar 4. Berbagai Jenis Board Mikrokontroler Arduino

Salah satu jenis mikrokontroler yang sangat umum digunakan bagi kalangan pemula adalah Arduino. Arduino merupakan board Mikrokontroler yang menggunakan chip AVR ATmega yang dirilis oleh Atmel dan bersifat open source. Adapun tipe-tipe arduino yakni Arduino Uno, Arduino Due, Arduino Mega, Arduino Nano, Arduino Leonardo, dan lain-lain Gambar berikut merupakan tampilan fisik beberapa jenis Arduino yang banyak digunakan.

2. Espressif (ESP)

Mikrokontroler jenis ESP ini umum digunakan dalam project yang berhubungan dengan Internet of Things karena keunggulannya yang dapat disambungkan pada WiFi. Series mikrokontroler Espressif yang umum dipasaran adalah ESP8266 dan ESP32.

a. ESP8266

ESP8266 merupakan modul wifi yang berfungsi sebagai perangkat tambahan mikrokontroler seperti Arduino agar dapat terhubung langsung dengan wifi dan membuat koneksi TCP/IP. Selain itu modul ini berbasis SOC (*Single on Circuit*) yang menjadikan perangkat ini dapat juga digunakan tanpa bantuan mikrokontroler lain. Modul ini membutuhkan daya sekitar 3.3V dan memiliki tiga mode wifi yaitu *Station*, *Access Point* dan *Both* (Keduanya). Modul ini juga dilengkapi dengan prosesor, memori dan GPIO dimana jumlah pin bergantung dengan jenis ESP8266 yang kita gunakan. Berikut merupakan

jenis jenis dari mikrokontroler ESP8266 series.



Gambar 5. Berbagai Macam Modul ESP8266

Pada gambar diatas dapat kalian lihat berbagai macam model dari modul ESP8266, mulai dari esp-01 sampai dengan esp-14. Semua seri diatas hanyalah modul saja dan tidak dilengkapi dengan *board* tambahan sehingga kita harus membuat *board*-nya sendiri. Namun pada *onlineshop* atau di toko toko elektronik modul tersebut sudah tergabung dengan boardnya sehingga terkadang antara bentuk mikrokontroler ESP32 dan ESP8266 hanya sedikit perbedaan.



Gambar 6. Board NodeMCU ESP8266

b. ESP32

ESP32 adalah mikrokontroler yang dikenalkan oleh Espressif System dan merupakan penerus dari mikrokontroler ESP8266. Pada mikrokontroler ini sudah tersedia modul wifi dalam chip sehingga sangat mendukung untuk membuat sistem aplikasi *Internet of Things*. Terlihat pada gambar di bawah merupakan pin out dari ESP32. Pin tersebut dapat dijadikan input atau output untuk menyambungkan dengan aktuator maupun sensor lain sesuai dengan kebutuhan.



Gambar 7. Board ESP32

3. STMicroelectronics (STM)

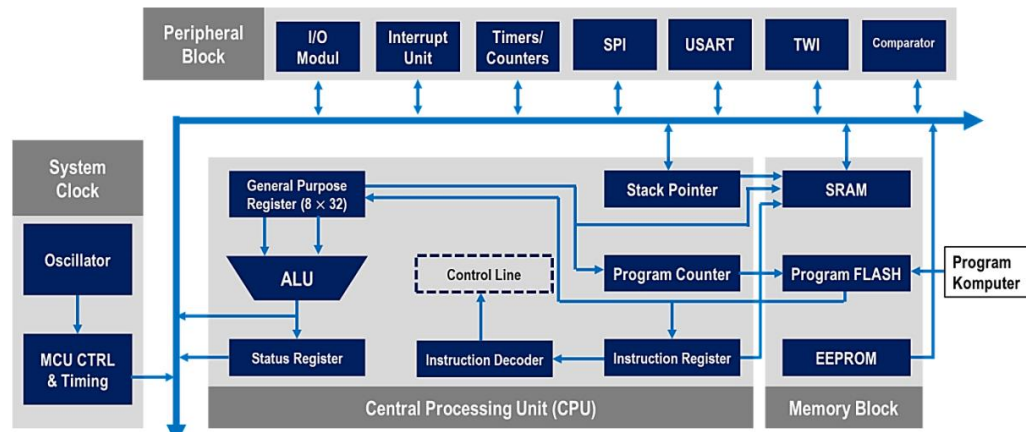
Mikrokontroler jenis STM umum digunakan pada proyek yang memerlukan kekuatan pemrosesan yang tinggi, salah satu seri mikrokontroler dari STMicroelectronics adalah STM32. Seri STM32 didasarkan pada inti ARM Cortex dengan seri M0, M0+, M3, M4, M7, dan M33 yang dirancang khusus untuk aplikasi tertanam yang memerlukan kinerja tinggi, biaya rendah, dan konsumsi daya rendah.



Gambar 8. STM32F103C8T6 “Blue Pill”

1.3. Arsitektur Microcontroller

Berikut merupakan diagram arsitektur mikrokontroler AVR.



Gambar 9. Diagram Arsitektur Mikrokontroler AVR

Fungsi dari masing-masing blok pada gambar tersebut adalah sebagai berikut :

1. CPU bertugas untuk memproses semua instruksi atau program yang ada di Mikrokontroler
2. Memory Block bertugas menyimpan data dan program. Program flash berfungsi untuk menyimpan kode program, sedangkan SRAM dan EEPROM berfungsi menyimpan data.
3. System Clock bertugas menangani semua clock pada CPU, memori, ADC, input/output, dan lain-lain.
4. Peripheral Block berfungsi untuk menghubungkan Mikrokontroler dengan environment di luar Mikrokontroler seperti I/O Modul, Interrupt Unit, Timer/Counter, SPI, USART, TWI, dan Comparator.

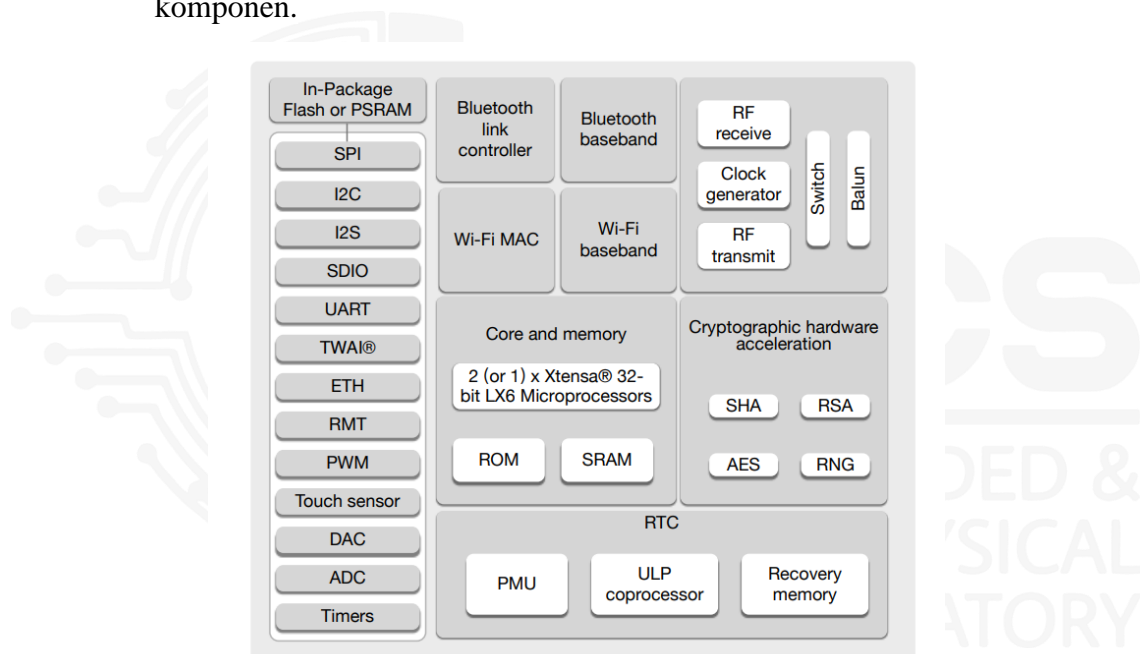
1.4. Spesifikasi Mikrokontroler

1. Datasheet

Spesifikasi lengkap setiap mikrokontroler dapat ditemukan dalam datasheet yang disediakan oleh produsennya. Datasheet mikrokontroler adalah dokumen teknis komprehensif yang memberikan informasi mendalam tentang berbagai aspek dari suatu mikrokontroler tertentu, termasuk spesifikasi teknis, fitur, dan fungsionalitasnya. Berikut adalah beberapa informasi yang dapat ditemukan dalam datasheet mikrokontroler.

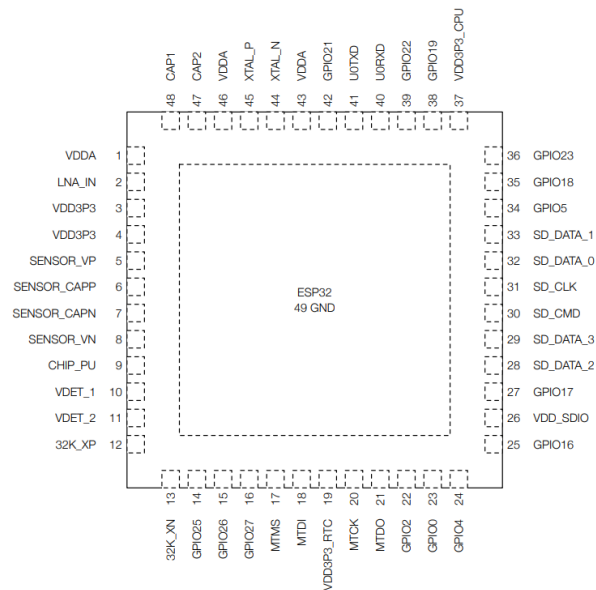
- a. Spesifikasi Teknis: Datasheet akan mencantumkan spesifikasi teknis seperti

- kecepatan clock, jumlah pin I/O, kapasitas RAM dan Flash, dan berbagai parameter lainnya yang berkaitan dengan kinerja perangkat.
- Arsitektur:** Datasheet akan menjelaskan arsitektur mikrokontroler, termasuk jumlah dan jenis core CPU yang digunakan, seperti core Xtensa, serta informasi tentang pipelining dan cache.
 - Modul Peripheral:** Ini mencakup daftar dan deskripsi modul periferifal yang ada di ESP32, seperti GPIO, UART, SPI, I2C, WiFi, Bluetooth, dan lainnya. Informasi ini mencakup pinout, fungsi, dan konfigurasi.
 - Diagram Blok:** Datasheet biasanya akan menyertakan diagram blok yang memperlihatkan struktur internal perangkat dan hubungan antara berbagai komponen.



Gambar 10. Diagram Blok pada Datasheet Mikrokontroler

- Diagram Pinout:** Datasheet biasanya menyertakan diagram pinout yang menunjukkan fungsi setiap pin, sehingga kita dapat merencanakan koneksi perangkat ketika akan mendesain suatu alat.



Gambar 11. Diagram Pinout pada Datasheet Mikrokontroler

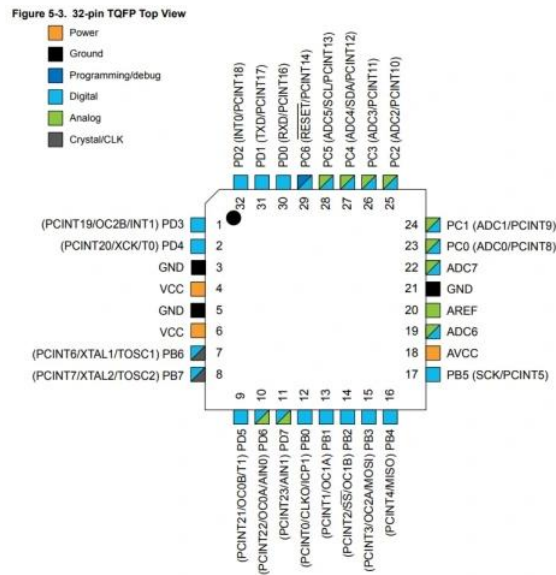
- f. **Konsumsi Energi:** Ini mencakup informasi tentang konsumsi daya perangkat dalam berbagai mode operasi, yang berguna untuk perencanaan daya dalam aplikasi baterai.

Parameter	Description	Min	Typ	Max	Unit
VDDA, VDD3P3_RTC ^{note 1} , VDD3P3, VDD_SDIO (3.3 V mode) ^{note 2}	Voltage applied to power supply pins per power domain	2.3/3.0 ^{note 3}	3.3	3.6	V
VDD3P3_CPU	Voltage applied to power supply pin	1.8	3.3	3.6	V
I_{VDD}	Current delivered by external power supply	0.5	—	—	A
T ^{note 4}	Operating temperature	-40	—	125	°C

Gambar 12. Tabel Konsumsi Energi pada Datasheet

2. Pinout

Mikrokontroler biasanya hadir dalam berbagai bentuk fisik, atau paket. Penomoran pin pada mikrokontroler dimulai dari sudut kiri atas, yang biasanya ditandai dengan titik. Dari situ, dapat dihitung sekitar meja chip searah jarum jam.

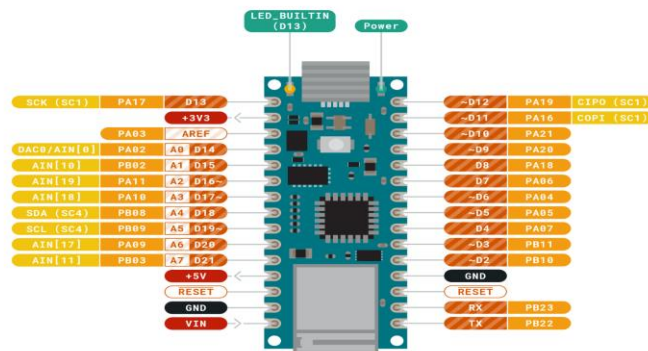


Gambar 13. Pinout Mikrokontroler

Untuk modul Mikrokontroler, penomoran ini tidak berlaku, karena papan tersebut memiliki beberapa header pin. Header pin biasanya diberi nomor, dan pin dari setiap header dihitung. Sayangnya, penomoran header tidak selalu mengikuti pola yang sama dengan penomoran IC. Pin mikrokontroler disebut dengan pin fisiknya (di mana mereka berada secara fisik di papan) dan nama pin fungsionalnya. Misalnya, pin fisik 20 pada Arduino Nano adalah pin I/O digital 2.



**ARDUINO
NANO 33 IoT**



Gambar 14. Pinout Modul Mikrokontroler Arduino Nano 33 IoT

BAB II

Konsep Dasar Sinyal Digital and Analog

2.1. Sinyal Digital vs Analog

Pada perkembangan dunia industri, sistem analog lebih dahulu dikenal sebelum adanya sistem digital. Sistem analog yaitu sistem yang nilai besarannya memiliki karakteristik kontinu. Contoh sistem analog yaitu seperti amplifier audio dan termometer merkuri. Seiring perkembangan zaman, ditemukan sistem Digital. Sistem digital yaitu sistem yang nilai besaran keluarannya bernilai antara 1 atau 0 (HIGH or LOW).

Sistem digital sering dipakai sampai sekarang karena lebih akurat dan lebih stabil apabila terkena noise. Sistem digital juga diterapkan pada elektronika yang lebih dikenal sebagai elektronika digital. Penentuan nilai 1 atau 0 pada elektronika digital berdasarkan dari tegangan. Jika tegangan berada pada nilai 2V sampai 5V maka akan bernilai 1 (HIGH). Jika tegangan berada pada nilai 0V sampai 0,8V maka akan bernilai 0 (LOW).

2.2. Bilangan Digital

Bilangan digital adalah representasi angka atau karakter dalam bentuk simbol-simbol digital atau kode-kode tertentu. Berikut merupakan beberapa jenis bilangan digital:

1. Desimal

Sistem ini merupakan sistem bilangan yang umum digunakan dalam komunikasi sehari-hari. Sistem bilangan ini mempunyai 10 simbol yaitu dari angka 0 hingga 9. Setiap kedudukan angka mempunyai nilai kelipatan 10, mulai dari angka paling kanan yakni dari $10^0, 10^1, 10^2, 10^3$ dst. Setiap kedudukan mempunyai besaran tertentu yang secara urut dimulai dari kanan disebut satuan, puluhan, ratusan, dan ribuan dst.

	ribuan	ratusan	puluhan	satuan
	10^3	10^2	10^1	10^0

contoh :

Angka Desimal 10932 (10932₍₁₀₎)

1 0 9 3 2

Pertama	2 . 10 ⁰ = 2 .	1 =	2
Kedua	3 . 10 ¹ = 3 .	10 =	30
Ketiga	9 . 10 ² = 9 .	100 =	900
Keempat	0 . 10 ³ = 0 .	1000 =	0
Kelima	1 . 10 ⁴ = 1 .	10000 =	10000
			<hr/>
			10932

2. Biner

Sistem ini berbasis 2 dimana banyak digunakan untuk sinyal elektronik dan pemrosesan data. Sistem biner hanya mempunyai 2 simbol yang berbeda, sehingga pada sistem ini hanya dikenal angka “ 0 “ dan angka “1 “. Setiap kedudukan angka mempunyai nilai kelipatan 2, mulai angka paling kanan yakni dari 2⁰, 2¹, 2², 2³ dan seterusnya.

3. Hexadesimal

Sistem bilangan ini biasa digunakan untuk mempersingkat penulisan bilangan biner karena ia berbasis pada angka 16 sehingga mempunyai 16 simbol yang terdiri dari 10 angka dan 6 huruf yang dipakai pada sistem desimal yaitu adalah sebagai berikut 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Dan dari ke enam huruf tersebut mempunyai nilai sebagai berikut A= 10, B= 11, C=12, D=13, E= 14, F=15. Setiap kedudukan angka atau simbol heksadesimal memiliki nilai kelipatan 16, mulai dari kanan yakni 16⁰, 16¹, 16², 16³ dst.

4. ASCII (American Standard Code for Information Interchange)

ASCII merupakan sebuah standar internasional dalam kode huruf dan simbol seperti Hex dan Unicode. Kode ASCII memiliki komposisi bilangan biner sebanyak 8 bit. Dimulai dari 00000000 hingga 11111111. Total kombinasi yang dihasilkan sebanyak 256, dimulai dari kode 0 hingga 255 dalam sistem bilangan Desimal. Pada dasarnya ASCII dapat mewakili angka, huruf, maupun karakter-karakter lainnya di dalam komputer, misalnya 1,2,3,A,B,@,#,\$, dan karakter lainnya. Karakter ini selalu digunakan oleh komputer dan alat komunikasi lain untuk menunjukkan teks (String). Sebagai contohnya adalah karakter 'A' memiliki kode ASCII 65 dalam bilangan desimal, 1000001 dalam biner, dan 41 dalam heksadesimal.

2.3. Konversi Bilangan Digital

1. Konversi Bilangan Desimal ke Bilangan Biner

Contoh :

Konversi Bilangan Desimal $Z_{(10)} = 83$ ke bilangan Biner $Z_{(2)}$. Angka 83 dibagi dengan basis bilangan biner yaitu 2.

$$83 : 2 = 41 \quad (\text{Sisa } 1)$$

Sisa 1 ini merupakan digit pertama dari bilangan biner ...x x x x 1. Untuk mendapatkan harga pada digit berikutnya adalah :

$$41 : 2 = 20 \quad (\text{Sisa } 1)$$

Sisa 1 ini merupakan digit pertama dari bilangan biner ...x x x 1 1. Untuk mendapatkan harga pada digit berikutnya adalah :

83	:	2	=	41	sisanya	1	
41	:	2	=	20	sisanya	1	
20	:	2	=	10	sisanya	0	
10	:	2	=	5	sisanya	0	
5	:	2	=	2	sisanya	1	
2	:	2	=	1	sisanya	0	
1	:	2	=	0	sisanya	1	

$83_{(10)} = 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1_{(2)}$

Jadi $Z_{(10)} = 83$ adalah $Z_{(2)} = 1010011$.

2. Konversi Bilangan Desimal ke Bilangan Hexadesimal

Contoh :

Konversi Bilangan Desimal $Z_{(10)} = 10846$ ke bilangan Heksadesimal $Z_{(16)}$

10846	:	16	=	677	sisanya	14	
677	:	16	=	42	sisanya	5	
42	:	16	=	2	sisanya	10	
2	:	16	=	0	sisanya	2	

$10846_{(10)} = 2 \ A \ 5 \ E_{(16)}$

Jadi $Z_{(10)} = 10846$ adalah $Z_{(16)} = 2A5E$.

3. Konversi Bilangan Biner ke Bilangan Desimal

Contoh :

Konversi Bilangan Biner $Z_{(2)} = 10101010$ ke bilangan Desimal $Z_{(10)}$

1 0 1 0 1 0 1 0	
0 . 2^0 = 0 . 1 = 0	
1 . 2^1 = 1 . 2 = 2	
0 . 2^2 = 0 . 4 = 0	
1 . 2^3 = 1 . 8 = 8	
0 . 2^4 = 0 . 16 = 0	
1 . 2^5 = 1 . 32 = 32	
0 . 2^6 = 0 . 64 = 0	
1 . 2^7 = 1 . 128 = 128	
<hr/>	
$10101010_{(2)} = 170_{(10)}$	

Jadi $Z_{(2)} = 10101010$ adalah $Z_{(10)} = 170$

4. Konversi Bilangan Biner ke Bilangan Hexadesimal

Contoh :

Konversi Bilangan Biner $Z_{(2)} = 101101$ ke bilangan Hexadesimal $Z_{(16)}$

Langkah pertama : Konversikan bilangan biner ke bilangan desimal

1 0 1 1 0 1	
1 . 2^0 = 1 . 1 = 1	
0 . 2^1 = 0 . 2 = 0	
1 . 2^2 = 1 . 4 = 4	
1 . 2^3 = 1 . 8 = 8	
0 . 2^4 = 0 . 16 = 0	
1 . 2^5 = 1 . 32 = 32	
<hr/>	
$101101_{(2)} = 45_{(10)}$	

Langkah Kedua : Konversikan bilangan desimal ke hexadesimal

45 : 16 = 2	sisanya 13	
2 : 16 = 0	sisanya 2	
<hr/>		
$45_{(10)} = 2D_{(16)}$		

Jadi $Z_{(2)} = 101101$ adalah $Z_{(16)} = 2D$

5. Konversi Bilangan Hexadesimal ke Bilangan Desimal

Contoh :

Konversi Bilangan Heksadesimal $Z_{(16)} = B3C9$ ke bilangan Desimal $Z_{(10)}$

B 3 C 9

$$\begin{array}{rcl}
 9 \cdot 16^0 & = & 9 \cdot 1 = 9 \\
 12 \cdot 16^1 & = & 12 \cdot 16 = 192 \\
 3 \cdot 16^2 & = & 3 \cdot 256 = 768 \\
 11 \cdot 16^3 & = & 11 \cdot 4096 = 45056 \\
 \hline
 \text{B3C9}_{(16)} & = & 46025_{(10)}
 \end{array}$$

Jadi $Z_{(16)} = \text{B3C9}$ adalah $Z_{(10)} = 46025$

6. Konversi Bilangan Hexadesimal ke Bilangan Biner

Contoh :

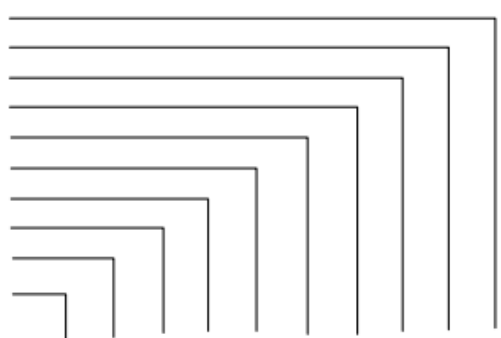
Konversi Bilangan Heksadesimal $Z_{(16)} = 2\text{FC}$ ke bilangan Biner $Z_{(2)}$

Langkah pertama

2 F C

$$\begin{array}{rcl}
 12 \cdot 16^0 & = & 12 \cdot 1 = 12 \\
 15 \cdot 16^1 & = & 15 \cdot 16 = 240 \\
 2 \cdot 16^2 & = & 2 \cdot 256 = 512 \\
 \hline
 2\text{FC}_{(16)} & = & 764_{(10)}
 \end{array}$$

Langkah Kedua

764	:	2	=	382	sis	0	
382	:	2	=	191	sis	0	
191	:	2	=	95	sis	1	
95	:	2	=	47	sis	1	
47	:	2	=	23	sis	1	
23	:	2	=	11	sis	1	
11	:	2	=	5	sis	1	
5	:	2	=	2	sis	1	
2	:	2	=	1	sis	0	
1	:	2	=	0	sis	1	

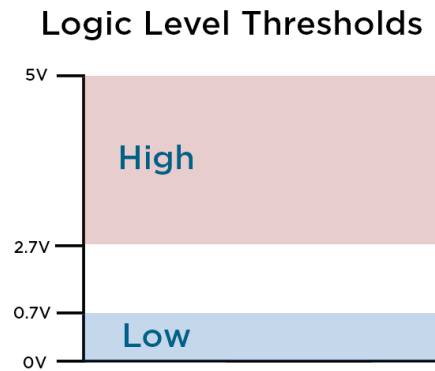
$764_{(10)} = 1\ 0\ 1\ 1\ 1\ 1\ 1\ 0\ 0_{(2)}$

Jadi $Z_{(16)} = 2\text{FC}$ adalah $Z_{(2)} = 1011111100$

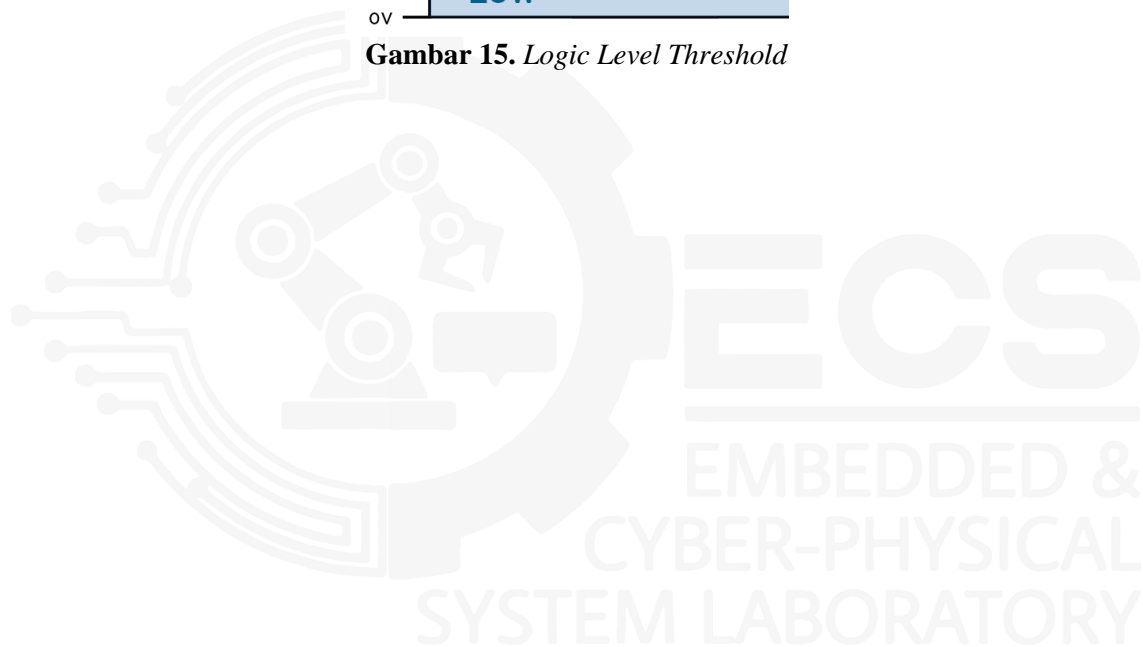
2.4. Tegangan Logic Level

Tingkat logika mengacu pada tegangan sinyal tertentu yang mewakili kondisi biner 0 atau 1 dalam rangkaian digital. Tingkat-tingkat ini sangat penting untuk fungsi sistem digital, karena akan mendefinisikan bagaimana perangkat menginterpretasikan sinyal listrik untuk melakukan operasi. Tingkat logika sangat penting dalam memastikan

komunikasi yang tepat antara berbagai komponen dari sistem elektronik. Interpretasi yang benar dari level-level ini memastikan bahwa sirkuit digital berfungsi dengan normal, karena setiap kesalahan interpretasi dapat menyebabkan malfungsi atau kesalahan data input/output dari sistem.



Gambar 15. *Logic Level Threshold*



BAB III

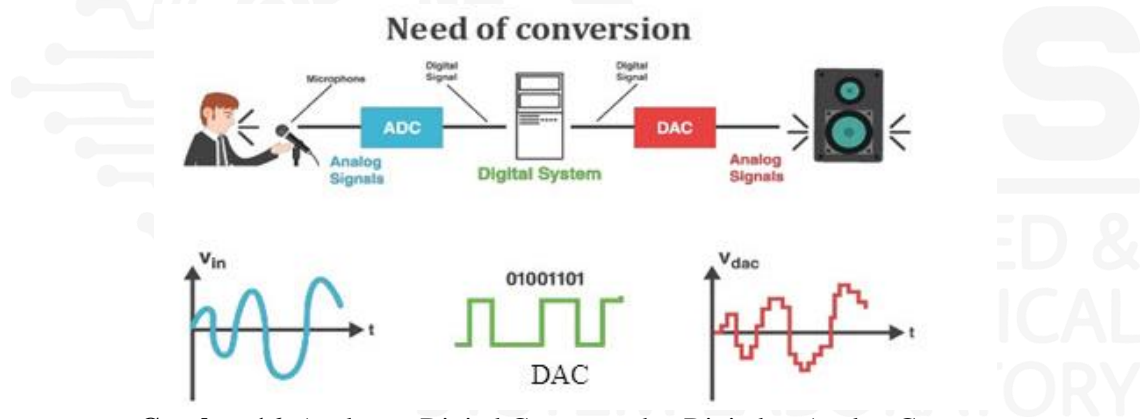
Fitur-Fitur Mikrokontroler

3.1. Analog to Digital Converter

ADC (Analog to Digital Converter) adalah perangkat elektronika yang berfungsi untuk mengubah sinyal analog (sinyal kontinu) menjadi sinyal digital. Perangkat ADC dapat berbentuk suatu modul atau rangkaian elektronika maupun suatu chip IC. ADC berfungsi untuk menjembatani pemrosesan sinyal analog oleh sistem digital.

3.2. Digital to Analog Converter

DAC (Digital to Analog Converter) adalah perangkat elektronika yang berfungsi untuk mengubah sinyal digital (diskrit) menjadi sinyal analog (kontinu). Aplikasi DAC adalah sebagai antarmuka (interface) antara perangkat yang bekerja dengan sistem digital dan perangkat pemroses sinyal analog. Perangkat DAC dapat berupa rangkaian elektronika dan chip IC DAC.



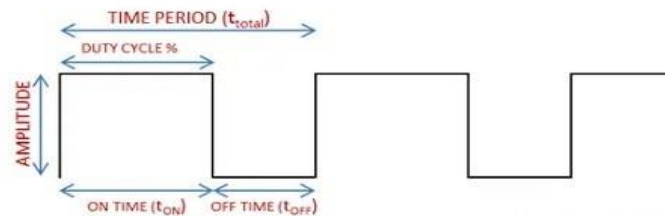
Gambar 16. Analog to Digital Converter dan Digital to Analog Converter

Konverter Digital ke Analog ini juga sering disingkat menjadi DAC, D2A atau D/2. Sinyal yang diubah menjadi bentuk Analog biasanya digunakan untuk menggerakkan motor, diafragma speaker, pengendali suhu dan lain sebagainya pada suatu rangkaian atau perangkat elektronika.

3.3. Pulse Width Modulation (PWM)

PWM memiliki cara kerja yang berbanding terbalik dengan ADC (Analog Digital Converter) yaitu untuk menghasilkan sinyal analog dari perangkat digital. PWM (Pulse Width Modulation) sering disebut sebagai modulator lebar pulsa. Fungsi PWM adalah sebagai metode yang digunakan untuk memanipulasi lebar pulsa yang terdapat pada

sebuah gelombang kotak. Untuk membangkitkan sinyal PWM, ada beberapa cara yang dapat dilakukan diantaranya dengan menggunakan mikrokontroler seperti AVR maupun Arduino.



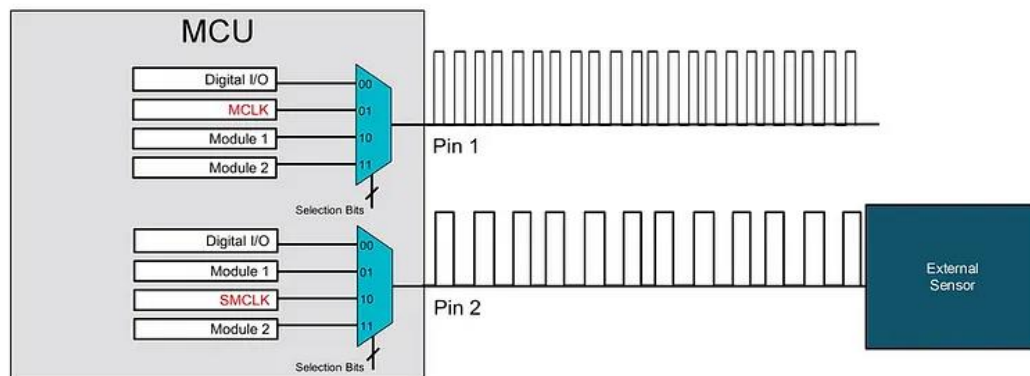
Gambar 17. *Pulse Width Modulation*

Siklus kerja PWM pada umumnya sinyal PWM akan tetap dalam pada posisi ON (High) untuk waktu yang ditentukan, kemudian akan OFF (Low) selama sisa periodenya. Sebagai pengguna, kita dapat menentukan berapa lama PWM berada dalam posisi ON. Caranya yaitu dengan mengendalikan siklus kerja (duty cycle) dari PWM. Pada saat PWM dalam posisi ON, siklus kerja atau study cycle memiliki nilai 100%. Sedangkan pada saat PWM OFF, disebut juga PWM dalam posisi duty cycle 0%. Penerapan PWM biasanya ditemukan pada beberapa situasi. Misalnya digunakan untuk mengatur kecepatan motor DC, mengatur redup dan cerahnya LED, pengendalian sudut motor servo dan lain sebagainya.

3.4. Timing & Clock

Clock memberikan pulsa dalam sistem elektronik. Dalam rantai sinyal, clock adalah penyedia waktu sampel referensi untuk proses konversi ADC, DAC dan media antarmuka lainnya. Clock diperlukan dalam sistem mikrokontroler yang memerlukan waktu referensi untuk sinkronisasi, eksekusi perintah, dan transfer data. Menyediakan input frekuensi ke berbagai perangkat, memungkinkan perangkat tersebut untuk menjalankan fungsi yang dimaksudkan

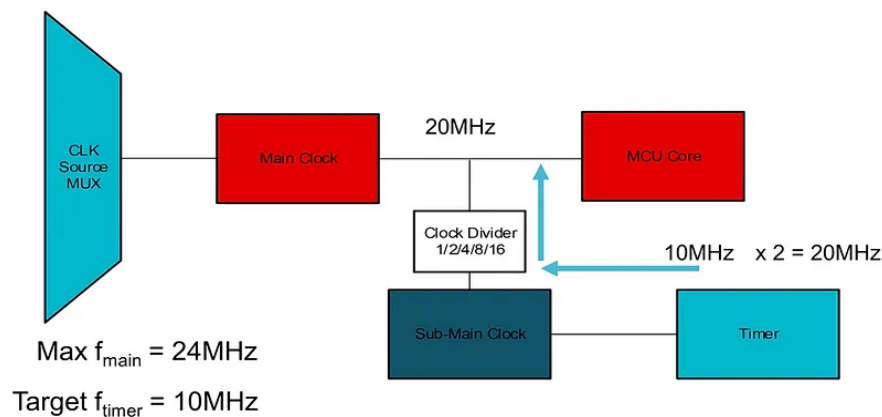
Outputting a clock within an MCU



Gambar 18. Clock pada Mikrokontroler

Sebagian besar MCU memiliki kemampuan untuk menghasilkan sinyal clock pada pin tertentu. Untuk mengaktifkan fitur ini, perlu dilakukan konfigurasi pin GPIO (General Purpose Input/Output). GPIO sering kali akan memiliki banyak fungsi, maka akan dihubungkan dengan sebuah divais pemilihan (*multi input –single output*) yang dikenal dengan *multiplexer*. Dalam contoh ini, perlu konfigurasi MUX pin 1 ke nilai 01 dan MUX pin dua ke nilai 10 untuk menghasilkan keluaran MCLK dan SMCLK masing-masing.

Frequency planning



Gambar 19. Frequency Planning pada sistem Mikrokontroler

Ada beberapa alasan diperlukannya frekuensi clock ke sebuah pin. Salah satunya adalah untuk menggunakan clock sebagai sumber eksternal bagi komponen lain dalam sistem. Perlu diperhatikan kemampuan dan fitur pin MCU karena dapat mempengaruhi cara mengarahkan atau menyediakan clock ke komponen sistem lainnya. Alasan lainnya adalah untuk keperluan kalibrasi atau debugging sistem, di mana sinyal clock digunakan untuk memastikan sinkronisasi dan fungsi sistem berjalan sesuai harapan.

BAB IV

Serial Komunikasi Mikrokontroler

4.1. Komponen Sistem Komunikasi

1. Pengirim

Dalam sistem mikrokontroler merupakan bagian yang bertanggung jawab untuk mengolah dan mengubah data digital menjadi sinyal yang sesuai untuk transmisi. Proses ini melibatkan beberapa tahap:

- Persiapan data: Data diformat sesuai protokol komunikasi yang digunakan (seperti UART, SPI, atau I2C)
- Encoding: Data di encoding untuk transmisi yang efisien dan deteksi kesalahan
- Modulasi: Jika diperlukan, sinyal dimodulasi untuk transmisi (terutama untuk komunikasi nirkabel)
- Penguatan: Sinyal dikuatkan sesuai kebutuhan medium transmisi

2. Medium Transmisi

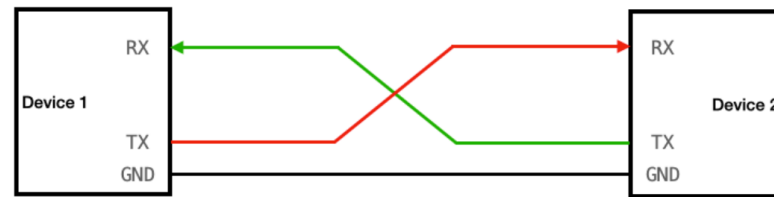
Media penghantar sinyal komunikasi, sebagai jalan untuk sinyal dari pengirim menuju penerima.

3. Penerima

Menerima sinyal transmisi dari pengirim yang akan diterjemahkan lagi menjadi sinyal lain atau menjadi instruksi/perintah tertentu. bisa juga menjadi data yang akan disimpan.

4.2. Universal Asynchronous Receiver/Transmitter

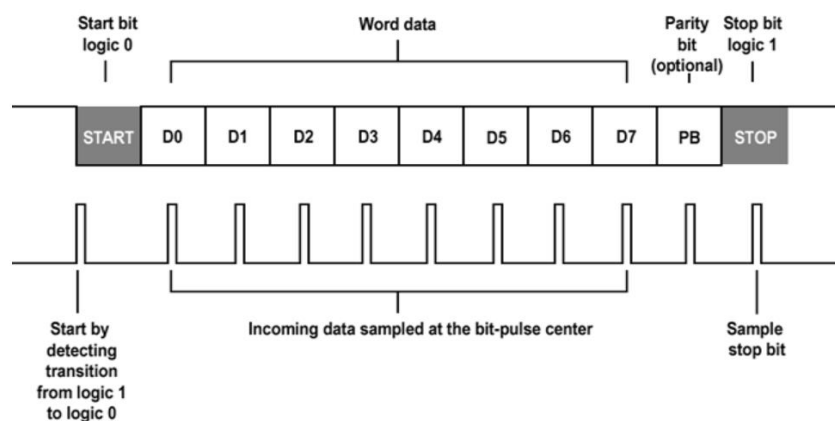
UART (*Universal Asynchronous Receiver/Transmitter*) merupakan protokol komunikasi serial yang memungkinkan perangkat untuk mengirim dan menerima data per bit secara asinkron. Artinya, data dikirimkan dalam bentuk urutan bit tanpa sinyal clock eksternal. Setiap perangkat memiliki tempat penyimpanan sementara (buffer) untuk menampung data yang dikirim dan diterima. Kecepatan komunikasi ditentukan oleh baud rate, yang mewakili jumlah transmisi sinyal yang terjadi dalam satu detik



Gambar 20. Skema Komunikasi UART

Untuk mengirim sebuah *byte* secara asinkron, *byte* tersebut akan diletakkan pada sebuah paket data yang diawali dengan *start bit*. *Start bit* menandakan bahwa data akan segera dikirimkan dan memicu proses sinkronisasi clock antara penerima dengan pengirim agar dapat membaca paket data yang akan dikirim dengan benar.

Setelah *start bit* dikirim, setiap bit yang menyusun data *byte* akan dikirim satu persatu dengan urutan *Least Significant Bit* (LSB) terlebih dahulu. Setiap bit dikirimkan dengan periode yang sama dengan setiap bit yang lain. Penerima akan menerjemahkan bit yang diterima dengan mengukur nilai tegangan pada titik tengah pulsa untuk menentukan bit tersebut bernilai *low* atau *high*, lalu mengumpulkannya untuk membentuk sebuah data *byte* yang lengkap.



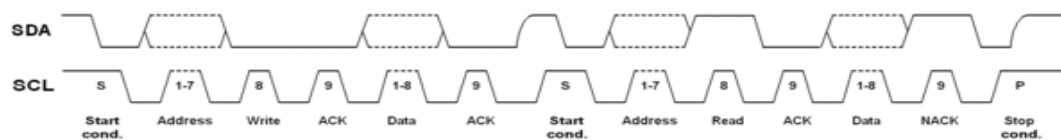
Gambar 21. Format Paket Data pada Komunikasi UART

Bit berikutnya yang dimasukkan ke dalam paket data adalah bit paritas, yang digunakan oleh penerima untuk melakukan pengecekan kesalahan sederhana. Bit paritas ini membantu memastikan bahwa jumlah bit yang bernilai benar (HIGH) sesuai dengan aturan paritas yang telah disepakati. Jika jumlah bit paket data yang bernilai benar (HIGH) ada genap, maka bit paritas akan bernilai 0 dan berlaku sebaliknya. Pada bagian akhir dari paket diletakkan *stop bit*. Setelah menerima seluruh paket data, penerima akan memeriksa bit paritas untuk mendeteksi kesalahan, kemudian memastikan adanya stop

bit untuk menyelesaikan proses komunikasi informasi.

4.3. Inter Integrated Circuit

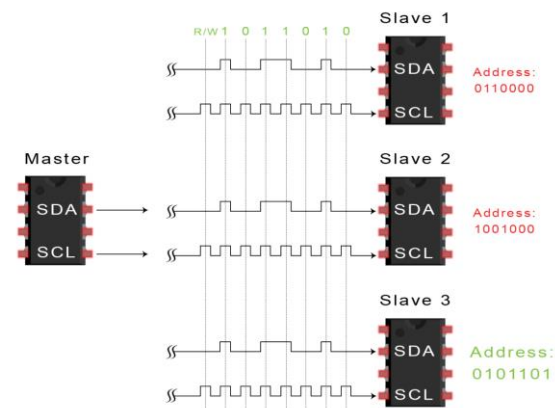
I2C (*I-squared-C*) atau *Inter Integrated Circuit* adalah serial komunikasi dua arah yang dapat mengirim dan menerima data. I2C memiliki dua saluran, yaitu *Serial Clock* (SCL) dan *Serial Data* (SDA). External device yang dihubungkan dengan I2C bisa menjadi master atau slave dimana master merupakan device yang memulai transfer dengan membentuk sinyal start dan mengakhirinya dengan membentuk sinyal stop. Sementara itu, slave merupakan device yang diatur oleh master.



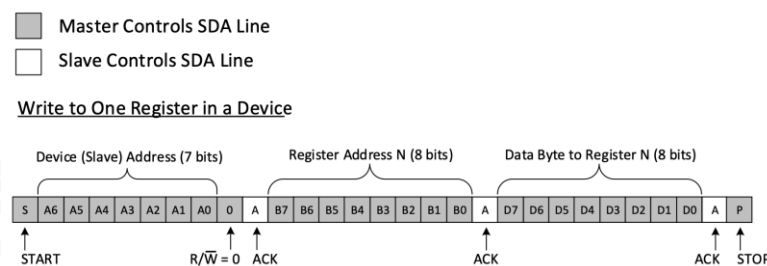
Gambar 22. Diagram Waktu Protokol Komunikasi I2C

Serial Data (SDA) adalah jalur di mana informasi (data, *address*, dan *acknowledgement*) dikirimkan antara perangkat master dan slave. Di sisi lain, *Serial Clock* (SCL) adalah jalur yang tugasnya mengatur sinkronisasi waktu komunikasi. Jalur ini memberikan sinyal kepada perangkat lain kapan harus mengirim atau menerima data. Jenis-jenis bit informasi yang dikirimkan meliputi:

1. Start (1 bit): bit ini menunjukkan bahwa perangkat master akan mengirimkan data. Sinyal ini ditandai dengan transisi *logic* dari tinggi ke rendah pada SDA saat SCL berada pada *logic* HIGH.
2. Address (7 bit): setelah kondisi start dimulai, perangkat master akan mengirimkan alamat perangkat slave yang ingin ditransmisi. Alamat ini terdiri dari 7 bit data. Contoh proses pemilihan address seperti pada Gambar 23.
3. Read/Write (1 bit): bit ini bernilai 0 jika master ingin menulis data ke slave (write) dan bernilai 1 jika master ingin membaca data dari slave (read);
4. ACK dan NACK (Acknowledge atau Not, 1 bit): ACK adalah sinyal yang menandakan bahwa slave telah membaca koneksi dari master dan siap untuk menerima data dari slave (write) atau mengirimkan data ke master (read)
5. Data (1 byte atau 8 bit): data transmisi I2C berupa 1 byte (8 bit). Data ini diisi oleh master jika pada mode write, dan diisi oleh slave jika pada mode read.

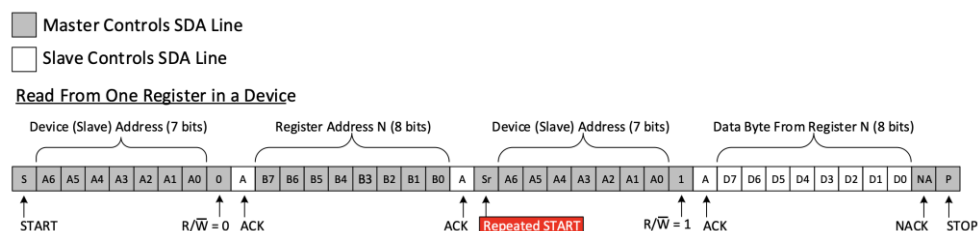


Gambar 23. Pemilihan Address pada protokol I2C



Gambar 24. Protokol I2C untuk Menulis Data Register

Mode transmisi I2C yang pertama adalah transmisi write. Mode transmisi write dimulai dengan sinyal start dari master. Selanjutnya, master mengirimkan alamat slave tujuan komunikasi, diikuti oleh bit bernilai 0 sebagai mode write. Setelah itu, slave membalas dengan ACK (bit 0) untuk mengonfirmasi bahwa alamatnya sesuai dan siap untuk menerima data dari master. Master kemudian mengirimkan alamat register (lokasi data) yang akan ditulis dan dibalas dengan sinyal ACK oleh slave. Setelah itu, master mengirimkan data 1 byte (8 bit) yang ingin ditulis ke register tersebut, dibalas dengan sinyal ACK oleh slave untuk menandakan bahwa transfer data berhasil. Proses write kemudian diakhiri dengan sinyal stop untuk menandakan bahwa komunikasi telah selesai.

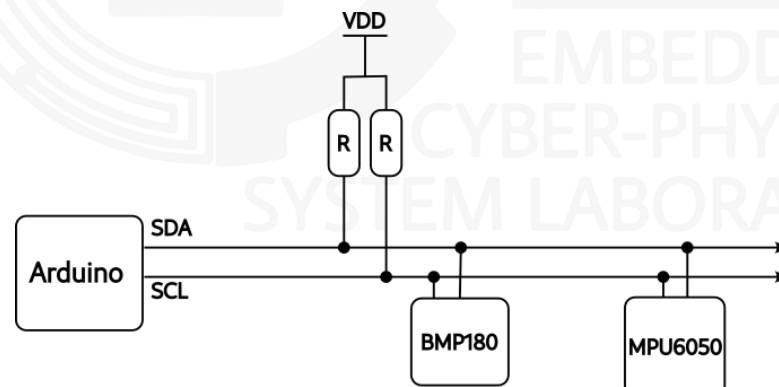


Gambar 25. Protokol I2C untuk Membaca Data Register

Proses transmisi mode read digunakan untuk membaca data dari register di dalam memori slave. Dimulai dengan bit start untuk menandakan bahwa master akan memulai

komunikasi. Selanjutnya, master mengirimkan alamat perangkat slave yang ingin dibaca diikuti dengan bit bernilai 0 untuk menandakan mode write (sengaja digunakan bit write untuk menuliskan alamat data terlebih dahulu). Kemudian dibalas dengan sinyal ACK oleh slave untuk mengonfirmasi bahwa alamat slave sesuai dan siap untuk melanjutkan komunikasi. Setelah itu, master mengirimkan alamat register milik data yang akan dibaca, dibalas dengan sinyal ACK dari slave sebagai konfirmasi bahwa alamat register tersebut benar.

Setelah itu, master mengirimkan sinyal *repeated start*. Perbedaan sinyal ini dengan start biasa adalah sinyal ini dikirimkan tepat setelah ACK dari slave, tanpa melalui proses stop terlebih dahulu. Barulah master mengirimkan alamat slave diikuti dengan bit bernilai 1 untuk menandakan mode read sesungguhnya. Kemudian slave membalas dengan sinyal ACK, dan master memberikan akses SDA kepada slave dengan mempertahankan jalur SCL untuk tetap menyuplai clock menuju slave. Slave mengirimkan 1 byte data (8 bit) menggunakan jalur SDA dan master membalas dengan ACK jika data sudah diterima. Proses ini berulang terus-menerus sampai ukuran data yang ditransmisikan terpenuhi. Kemudian master mengirimkan sinyal NACK agar slave menghentikan proses pengiriman data dan memberikan kembali akses SDA kepada master, diakhiri dengan kondisi stop.

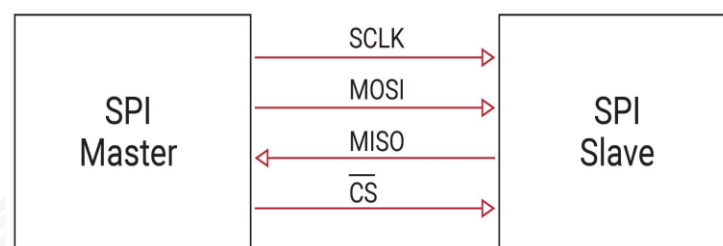


Gambar 26. Skema rangkaian I2C Arduino dengan sensor BMP180 dan MPU6050

4.4. Serial Peripheral Interface

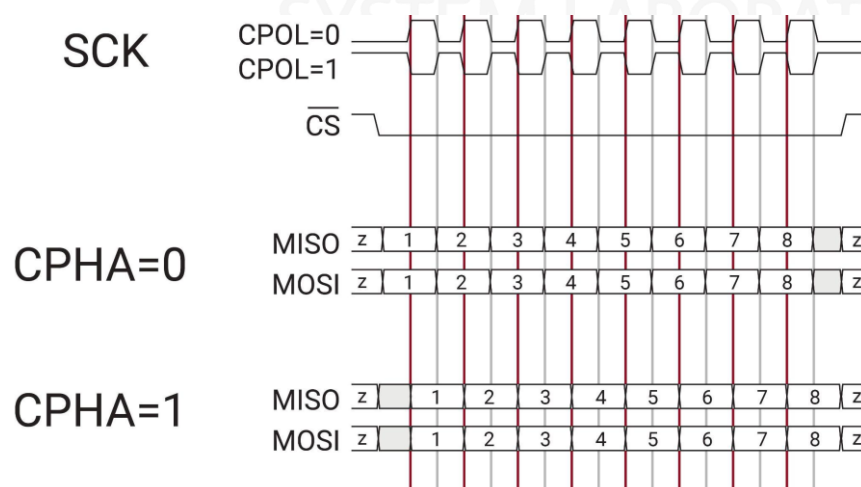
SPI (*Serial Peripheral Interface*) merupakan protokol komunikasi mikrokontroler yang menggunakan jalur clock bersama untuk sinkronisasi dan biasanya beroperasi pada laju bit melebihi 10 MHz. komunikasi SPI akan mengirim data antara beberapa perangkat. Perangkat-perangkat ini diatur dalam konfigurasi master dan slave, di mana master memiliki kontrol atas slave dan slave menerima instruksi dari master.

Implementasi SPI yang paling umum terdiri dari konfigurasi di mana satu perangkat berfungsi sebagai master, dan perangkat lainnya berfungsi sebagai slave. SPI adalah protokol komunikasi sinkron yang mentransmisikan dan menerima informasi secara bersamaan dengan kecepatan transfer data tinggi dan dirancang untuk komunikasi jarak pendek. Ada dua jalur data yang digunakan dalam komunikasi SPI yang dikenal sebagai MOSI dan MISO. Sinyal MOSI mengirimkan data dari master dan diterima oleh semua slave. Demikian pula, jalur data MISO mentransmisikan data dari salah satu perangkat slave ke perangkat master.



Gambar 27. Diagram Komunikasi SPI antara Master dan Slave

Sinyal clock dihasilkan oleh perangkat master dengan frekuensi tertentu dan digunakan untuk menyinkronkan data yang dikirim dan diterima antara perangkat. Sinyal ini dapat dikonfigurasi oleh master dengan menggunakan dua properti yang dikenal sebagai clock polarity (CPOL) dan clock phase (CPHA). Clock phase menentukan polaritas sinyal jam dan dapat dikonfigurasi untuk diam di level rendah (0) atau tinggi (1). Sinyal clock yang diam di posisi rendah memiliki pulsa tinggi dan tepi naik, sedangkan sinyal jam yang diam di posisi tinggi memiliki pulsa rendah dan tepi turun.



Gambar 28. Konfigurasi Clock Polarity (CPOL) dan Clock Phase (CPHA)

Clock phase menentukan waktu di mana data akan dimodifikasi dan dibaca. Jika

clock phase diatur ke nol, data dimodifikasi pada tepi trailing sinyal clock dan data dibaca pada tepi leading. Sebaliknya, jika properti ini diatur ke satu, data diubah pada tepi depan sinyal clock dan dibaca pada tepi belakang. Saat siklus clock berlangsung, data dikirim bit demi bit, secara bersamaan, melalui jalur MOSI dan MISO.

4.5. Wireless

Beberapa mikrokontroler khusus yang telah terintegrasi dengan modul wifi atau bluetooth bawaan atau mikrokontroler yang awalnya tidak memiliki fitur bawaan ini bisa mengimplementasikan protokol komunikasi wireless dengan menambahkan module eksternal ke mikrokontroler. Dimana terdapat beberapa protokol komunikasi wireless yang tersedia. Setiap protokol memiliki kelebihan dan kekurangan masing-masing dan satu protokol bisa lebih cocok daripada yang lain tergantung pada aplikasinya. Berikut merupakan beberapa contoh dari serial komunikasi wireless.

1. Bluetooth Low Energy (BLE)

BLE adalah varian Bluetooth yang difokuskan pada penghematan daya. Aplikasi utama BLE adalah transmisi jarak pendek dari sejumlah kecil data (low bandwidth). BLE mendukung tidak hanya komunikasi titik-ke-titik, tetapi juga mode siaran, dan jaringan mesh. Karena sifat-sifatnya, BLE cocok untuk diaplikasikan ke perangkat yang memerlukan menukar sejumlah kecil data secara berkala yang berjalan pada sel koin.

2. Bluetooth Classic

Bluetooth adalah standar teknologi wireless yang digunakan untuk bertukar data antara perangkat tetap dan bergerak dalam jarak pendek. Ini dioptimalkan untuk streaming data terus-menerus.

3. Wi-Fi

Ada beberapa cara dalam mengaplikasikan WiFi di sistem mikrokontroler salah satunya adalah HTTP request. Dengan HTTP request satu mikrokontroler bertindak sebagai server (titik akses Wifi) dan mikrokontroler lainnya bertindak sebagai klien. (Wifi station).

4. MQTT

Message Queuing Telemetry Transport atau MQTT adalah sistem publikasi dan langganan ringan dimana dapat menerbitkan dan menerima pesan sebagai klien. MQTT

adalah protokol pesan sederhana, dirancang untuk perangkat terbatas dengan bandwidth rendah.

Untuk menggunakan MQTT untuk bertukar data, diperlukan broker MQTT yang bertanggung jawab untuk menerima semua pesan, menyaring pesan-pesan tersebut, dan menerbitkan pesan ke semua klien yang berlangganan. MQTT merupakan protokol wireless yang sangat cocok untuk implementasi IoT dengan banyak perangkat.

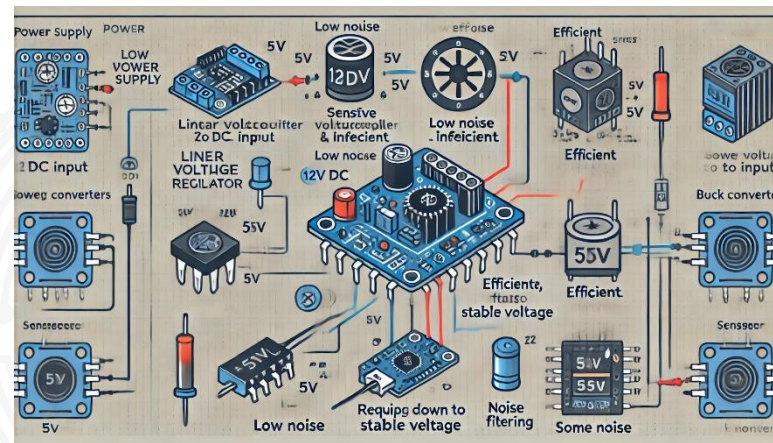


BAB V

Elektronika Daya pada Mikrokontroler

5.1. Pengenalan

Elektronika daya adalah cabang teknik listrik yang berfokus pada pengubahan dan pengendalian energi listrik. Dalam konteks mikrokontroler, elektronika daya sangat penting karena perangkat mikrokontroler ataupun sensor biasanya bekerja pada tegangan rendah, seperti 3.3V atau 5V, tetapi sering kali sumber dayanya lebih tinggi, seperti 12V atau 24V. Regulasi tegangan dan manajemen daya yang baik menjadi sangat penting untuk menjaga stabilitas, efisiensi, dan keandalan sistem.



Gambar 29. Elektonika Daya pada Mikrokontroler dan Sensor

5.2. Pengaplikasian Sumber tegangan DC

Mikrokontroler dan sensor biasanya bekerja pada tegangan yang cukup rendah, seperti 3.3V atau 5V, sementara sumber daya yang tersedia seringkali memiliki tegangan yang lebih tinggi, seperti 12V atau 24V. Oleh karena itu, regulasi tegangan menjadi sangat penting untuk memastikan tegangan yang diterima oleh mikrokontroler dan sensor tetap stabil dan tidak melebihi batas yang diizinkan. Dua jenis regulator yang sering digunakan adalah.

1. Linear Voltage Regulators (LDO - Low Dropout Regulators)

Regulasi linear sangat cocok untuk sensor yang sensitif terhadap noise karena memberikan output tegangan yang bersih dengan sedikit fluktuasi. Namun, LDO ini kurang efisien, terutama jika perbedaan antara tegangan input dan output sangat besar, karena energi yang terbuang sebagai panas. Contohnya, jika Anda mengubah 12V

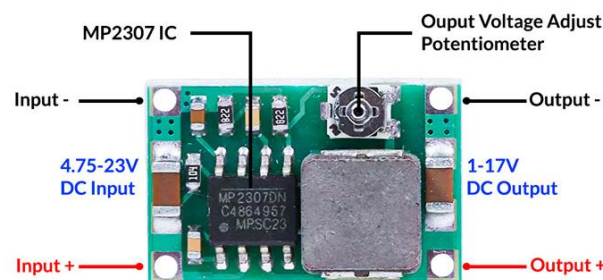
menjadi 3.3V, sebagian besar energi akan terbuang dalam bentuk panas.



Gambar 30. Macam-Macam Voltage Regulator

2. Switching Regulators

Regulator switching lebih efisien karena menggunakan teknik konversi switching untuk menurunkan tegangan. Meskipun lebih efisien, switching regulators bisa menimbulkan noise yang mungkin mengganggu sensor atau mikrokontroler yang sensitif, sehingga perlu menggunakan teknik filtrasi noise yang tepat.



Gambar 31. Contoh Variable Switching Regulator

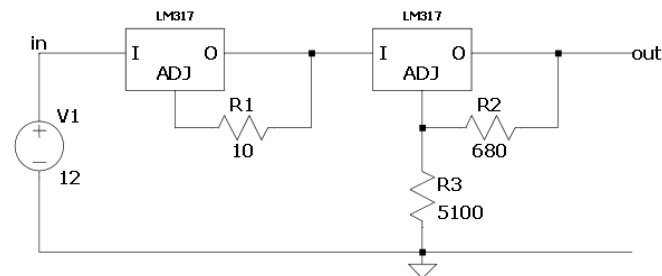
5.3. Pencegahan over-current pada mikrokontroler dan sensor

Over-current adalah kondisi di mana aliran arus listrik melebihi batas yang aman untuk sebuah komponen atau sirkuit. Kondisi ini bisa terjadi karena beberapa alasan, seperti hubungan pendek (short circuit), kegagalan perangkat, atau beban berlebih (overload). Setiap sirkuit atau komponen elektronik, seperti mikrokontroler atau sensor, dirancang untuk beroperasi dalam kisaran arus tertentu. Over-current dapat menyebabkan kerusakan fisik pada mikrokontroler dan sensor. Beberapa teknik untuk mencegah over-current dapat berupa.

1. Current Limiting Circuit

Dalam aplikasi tertentu, membatasi arus maksimum yang dapat ditarik oleh sensor atau mikrokontroler sangat penting untuk menghindari overheating. Dapat

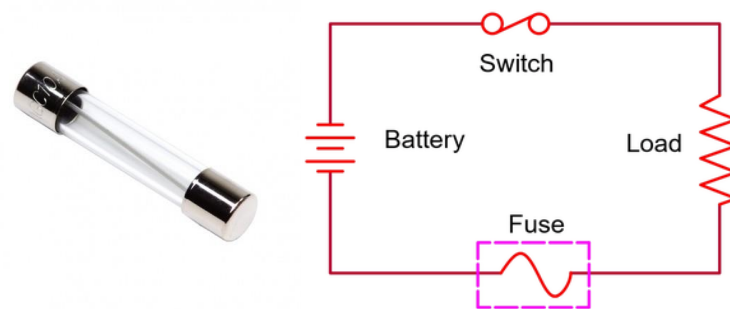
digunakan regulator tegangan dengan fitur pembatasan arus atau menambahkan rangkaian pembatas arus eksternal.



Gambar 32. Contoh Skematik Current Limiting Circuit

2. Fuse atau PTC Resettable Fuse

Komponen ini memiliki kapabilitas untuk memutuskan aliran listrik jika arus melebihi batas yang aman dan dapat digunakan untuk melindungi sirkuit dari kerusakan akibat over-current.



Gambar 33. Fuse dan Pengaplikasian di Rangkaian

5.4. Filtrasi Noise dan Proteksi Tegangan untuk Sensor

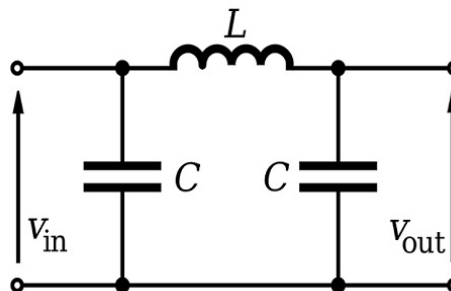
Sensor-sensor, terutama yang bekerja pada sinyal analog, sangat rentan terhadap noise dari switching power supply atau EMI (Electromagnetic Interference). Untuk mengatasi masalah ini, beberapa teknik yang umum digunakan adalah:

1. Capacitors

Menambahkan kapasitor bypass (decoupling) antara suplai daya dan ground dapat membantu mengurangi noise pada sensor. Misalnya, kapasitor keramik 0.1 μF sering digunakan pada pin daya mikrokontroler dan sensor.

2. Inductors

Induktor dapat membantu menyaring noise dari sumber daya switching. Mereka biasanya digunakan dalam kombinasi dengan kapasitor dalam konfigurasi filter LC untuk menghalangi frekuensi switching dari mencapai sensor atau mikrokontroler.



Gambar 34. Contoh Pengaplikasian Filter pada Skematik

5.5. Kesimpulan: Dos and Don'ts

Dos, hal-hal yang perlu dilakukan dan diperhatikan

1. Gunakan Regulator Tegangan yang Tepat: Pilih regulator tegangan, baik Linear Voltage Regulator (LDO) atau Switching Regulator (Buck Converter), yang sesuai untuk memastikan mikrokontroler dan sensor menerima tegangan yang stabil. Tegangan yang tidak stabil bisa menyebabkan malfungsi atau kegagalan fungsi pada mikrokontroler.
2. Implementasi Proteksi Over-current: Pastikan untuk menggunakan proteksi arus berlebih seperti fuse atau current-limiting circuits untuk mencegah kerusakan akibat over-current. Hal ini penting untuk melindungi komponen dari kerusakan akibat arus yang terlalu tinggi.
3. Gunakan Teknik Filtrasi Noise: Noise atau gangguan elektromagnetik bisa menyebabkan sensor bekerja tidak optimal. Gunakan komponen seperti kapasitor atau induktor di jalur daya untuk memfilter noise dan melindungi sinyal dari gangguan eksternal yang tidak diinginkan.

Don'ts, hal-hal yang harus dihindari

1. Jangan Biarkan Tegangan Input Terlalu Tinggi: Hindari membiarkan tegangan input jauh melebihi tegangan operasional tanpa regulator yang tepat. Misalnya, menggunakan sumber daya 12V pada mikrokontroler yang membutuhkan 5V tanpa

regulator bisa menyebabkan over-voltage, yang berpotensi merusak komponen.

2. Jangan Tempatkan Regulator Terlalu Dekat dengan Komponen Sensitif: Hindari menempatkan komponen regulator terlalu dekat dengan komponen sensitif, seperti sensor, tanpa adanya filtrasi noise. Hal ini bisa menyebabkan gangguan sinyal dan penurunan akurasi sensor.

Pentingnya Membaca Datasheet

Selalu baca datasheet komponen yang akan digunakan untuk memahami batas tegangan operasional dan arus yang dapat diterima. Setiap komponen memiliki spesifikasi yang berbeda, dan datasheet akan memberikan informasi kritis seperti toleransi tegangan, arus maksimum, dan kondisi operasi yang direkomendasikan. Memahami informasi ini sangat penting untuk mencegah kesalahan desain yang dapat menyebabkan kerusakan.

3.3.1 D.C. Electrical Characteristics

Typical Operating Circuit of section 4.2, VDD = 1.8V, VDDIO = 1.8V, T_A=25°C, unless otherwise noted.

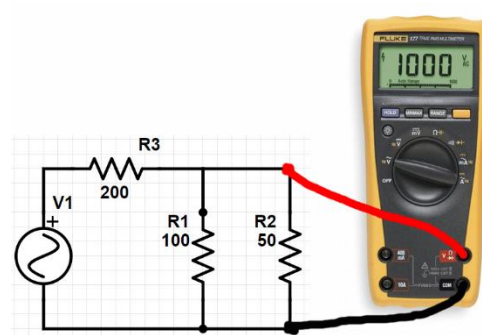
PARAMETER	CONDITIONS	MIN	TYP	MAX	Units	Notes
SUPPLY VOLTAGES						
VDD		1.71	1.8	3.45	V	1
VDDIO		1.71	1.8	3.45	V	1
SUPPLY CURRENTS						
Normal Mode	6-axis		3.4		mA	1
	3-axis Gyroscope		3.2		mA	1
	3-Axis Accelerometer, 4kHz ODR		450		μA	1
Accelerometer Low Power Mode	0.98 Hz update rate		7.27		μA	1, 2
	31.25 Hz update rate		18.65		μA	1, 2
Standby Mode			1.6		mA	1
Full-Chip Sleep Mode			6		μA	1
TEMPERATURE RANGE						
Specified Temperature Range	Performance parameters are not applicable beyond Specified Temperature Range	-40		+85	°C	1

Table 3: D.C. Electrical Characteristics

Gambar 35. Contoh Datasheet Electrical Characteristic Sensor

Pemeriksaan Tegangan Secara Isolasi

Sebelum memasang komponen pada rangkaian, lakukan pengujian tegangan secara isolasi. Ini berarti memeriksa tegangan output dari sumber daya dan regulator sebelum menghubungkannya dengan komponen. Langkah ini penting untuk memastikan tegangan yang diberikan sesuai dengan kebutuhan komponen dan untuk mencegah kesalahan seperti over-voltage yang bisa menyebabkan kerusakan.



Gambar 36. Pengukuran Tegangan menggunakan Multimeter

