



BUKU AJAR
Pengantar Teknik
Interfacing

PENULIS

Arief Wisaksono
Syamsudduha Syahrerini
Izza Anshory



Buku Ajar
Pengantar Teknik Interfacing

Oleh

Arief Wisaksono
Syamsudduha Syahririni
Izza Anshory



Diterbitkan oleh
UMSIDA PRESS
Jl. Mojopahit 666 B Sidoarjo
ISBN: 978-623-464-050-2
Copyright©2022
Authors
All rights reserved

Buku Ajar

Pengantar Teknik Interfacing

Penulis :

Arief Wisaksono

Syamsudduha Syahririni

Izza Anshory

ISBN :

978-623-464-050-2

Editor :

M.Tanzil Multazam,S.H.,M.Kn

Mahadika Darmawan,KW, .S.Pd,.M.Pd

Copy Editor :

Wiwit Wahyu Wijayanti

Design Sampul dan Tata Letak :

Wiwit Wahyu Wijayanti

Penerbit :

UMSIDA Press

Redaksi :

Universitas Muhammadiyah Sidoarjo

Jl. Mojopahit No 666B Sidoarjo, Jawa

Timur

Cetakan pertama, Agustus

2022

© Hak cipta dilindungi undang-undang

Dilarang memperbanyak karya tulis ini dengan suatu apapun
tanpa ijin tertulis dari penerbit.

DAFTAR ISI

	Cover dalam	1
	Daftar isi	4
	Tinjauan Mata kuliah	5
BAB I	Pengantar Antarmuka dan Peripheral dalam Sistem Komputer	8
BAB II	CPU dan Mikrokontroler	17
BAB III	Port I/O dan system clock Mikrokomputer	49
BAB IV	Sistem Minimum Mikrokontroler	60
BAB V	Timer,Counter, Interupsi dan real-time clock (RTC)	64
BAB VI	Antarmuka Digital Paralel	71
BAB VII	Antarmuka Serial	77
BAB VII	ADC dan DAC	86
BAB IX	Motor DC, Stepper dan Servo	90
BAB X	Modul Interfacing	126
	Modul Interfacing GPS Dengan Arduino UNO	126
	Modul Bluetooth HC-05 dengan Arduino UNO	129
	Interfacing Analog Joystick dengan Arduino UNO	131
	Interfacing LM35 Interfacing with Arduino UNO	133
	Antarmuka Motor DC dengan Arduino UNO	135
	Antarmuka Modul WiFi ESP8266 dengan Arduino UNO	138
	Daftar Pustaka	147

TINJAUAN MATA KULIAH

a. Deskripsi Singkat Mata Kuliah

Mahasiswa akan mempelajari tentang prinsip dan mekanisme kerja sistem Interface pada mikroprosesor, bagaimana system interface mikroprosesor dilakukan, kode program bahasa C++, dan eksekusi baris program. Akan dipelajari juga mengenai sinyal kendali dan antarmuka mikroprosesor dengan memori dan sistem I/O dalam pertukaran data. Diberikan juga materi mikrokontroler Arduino. Selain itu, diberikan juga materi tentang interfacing dari mikrokontroler Arduino

b. Kegunaan Mata Kuliah

Dengan mengikuti mata kuliah teknik interfacing pada mikroprosesor mahasiswa akan mendapatkan pengetahuan mengenai antar muka dan rangkaian kontrol dalam sebuah industri, dan mahasiswa mampu menjelaskan dan menentukan indikator-indikator yang berpengaruh terhadap rancang sisten kontrol, serta mahasiswa dapat mengimplementasikan rancang bangun system control.

c. Standar Kompetensi

Mahasiswa Mampu menerapkan dan mengembangkan teknik antarmuka, dalam sistem mikrokomputer sesuai dengan kebutuhan desain yang harus dipenuhi

3.1 Kompetensi Dasar

- a. Mahasiswa memahami rangkaian sifatnya. Antarmuka dan Peripheral dalam Sistem Komputer
- b. Mampu melakukan merancang dan melaksanakan eksperimen dalam membangun aplikasi mikrokontroller.
- c. Mampu menerapkan merencanakan dan membuat program mikrokontroller untuk berbagai aplikasi.
- d. Memahami komponen penyusun Sistem Minimum Mikrokontroler
- e. Memahami Antarmuka Digital Paralel
- f. Mahasiswa mampu menerapkan Antarmuka Serial
- g.
- h.

3.2 Indikator

- a. Mahasiswa akan mampu menjelaskan perbedaan antarmuka serial dan parallel
- b. Mahasiswa akan mampu menjelaskan perbedaan mikroprosesor, mikrokontroller dan mikrokomputer

- c. Mahasiswa akan mampu menjelaskan bus yang terdapat dalam mikrokomputer
- d. Mahasiswa akan mampu mengidentifikasi peripheral yang ada di satu sistem mikrokomputer dan antarmukanya
- e. Mahasiswa akan mampu mengaplikasikan rangkaian sumber daya yang stabil ke system mikrokomputer
- f. Mahasiswa akan mampu mengaplikasikan rangkaian clock ke sistem mikrokomputer sesuai dengan kebutuhan dan konstrain kecepatan mikrokomputer
- g. Mahasiswa akan mampu mengaplikasikan rangkaian reset ke system mikrokomputer dengan

h. Petunjuk Bagi Mahasiswa

Untuk membantu mahasiswa dalam mempelajari buku ajar pengantar mikroprocessor ini, dapat dilakukan beberapa tahapan sebagai berikut:

1. Bacalah dan pahami dengan teliti dari isi dan makna dari setiap bab. Kerjakan soal yang terdapat pada akhir setiap bab, kemudian lanjutkan ke bab selanjutnya.
2. Buku ajar ini disusun saling berkaitan antar bab, sehingga mahasiswa dalam mempelajari bab berikutnya harus memahami bab sebelumnya dengan baik.
3. Untuk membelajaran secara mandiri, disarankan mahasiswa dapat melengkapi dengan referensi lain yaitu buku teks yang berbeda namun materi yang disampaikan sama serta beberapa jurnal hasil penelitian.

PRAKATA

Alhamdulillahirobbil'aalamin, Segala puja dan puji syukur penulis panjatkan kepada Allah yang Maha Penyayang, tanpa karunia dan pertolongannya mustahil naskah buku ini terselesaikan, mengingat tugas dan kewajiban lain yang bersamaan hadir.

Buku ini ditulis bersamaan berdasarkan keinginan penulis yang berfungsi memberikan referensi untuk mahasiswa dalam mendesain peralatan kontrol sekaligus sebagai bahan untuk melatih ketrampilan dalam membuat device dalam system control.

Terselesainya penulisan buku ini tidak terlepas dari bantuan beberapa pihak, karena itu penulis menyampaikan terima kasih kepada dekan Fakultas Ilmu Kesehatan Universitas Muhammadiyah Sidoarjo yang telah memberi kesempatan untuk menulis buku. Penulis juga menyampaikan ucapan terima kasih kepada DRPM Universitas Muhammadiyah Sidoarjo atas kemudahan yang telah diberikan benar-benar bermanfaat bagi penulis untuk belajar menjadi pribadi yang lebih baik. Selain itu penulis juga menyampaikan rasa terima kasih kepada segenap pengelola Program Studi Teknik Elektro Universitas Muhammadiyah Sidoarjo untuk semua bantuannya.

Penulis menyadari bahwa buku ini masih mempunyai kelemahan sebagai kekurangannya, meskipun telah berusaha menghindarkan kesalahan, karena itu penulis berharap agar pembaca berkenan menyampaikan kritikan, dengan segala pengharapan dan keterbukaan penulis menyampaikan terima kasih dengan setulus-tulusnya. Kritik merupakan perhatian agar dapat menuju kesempurnaan.

Akhir kata, penulis berharap agar buku ini dapat membawa manfaat kepada pembaca. Secara khusus penulis berharap smoga buku ini dapat membantu mahasiswi berkomunikasi dan konseling dalam melaksanakan asuhan kebidanan sebagai salah satu tugas dan peran seorang Bidan.

Sidoarjo, Pebruari 2022

Penulis

BAB I

Pengantar Antarmuka dan Peripheral dalam Sistem Komputer

Mata Kuliah : TEKNIK INTERFACE

Capaian Pembelajaran :

Mahasiswa Mampu menerapkan dan mengembangkan teknik antarmuka, dalam sistem mikrokomputer sesuai dengan kebutuhan desain yang harus dipenuhi Kemampuan Akhir yang Diharapkan :

Mahasiswa memahami rangkaian sifatnya.

Antarmuka dan Peripheral dalam Sistem Komputer

Alokasi Waktu : 3 X 50 menit

Pertemuan ke : 1

Indikator :

- a) .Mahasiswa akan mampu menjelaskan perbedaan antarmuka serial dan parallel
- b) Mahasiswa akan mampu menjelaskan perbedaan mikroprosesor, mikrokontroller dan mikrokomputer
- c) Mahasiswa akan mampu menjelaskan bus yang terdapat dalam mikrokomputer
- d) Mahasiswa akan mampu mengidentifikasi peripheral yang ada di satu sistem mikrokomputer dan antarmukanya

Pengantar Antarmuka dan Peripheral dalam Sistem Komputer

Beberapa definisi akan dijelaskan dalam bagian ini sebelum membahas lebih mendalam mengenai apakah itu antarmuka (*interfacing*).

Definisi - definisi ini akan membentuk dasar pemahaman kita dan akan mempermudah pemahaman tentang *interfacing* selanjutnya.

Pemahaman yang diperlukan tersebut adalah ruang lingkup *interfacing* dan hubungan yang terkait dengannya.

- Beberapa definisi akan dijelaskan dalam bagian ini sebelum membahas lebih mendalam mengenai apakah itu antarmuka (*interfacing*).
- Definisi - definisi ini akan membentuk dasar pemahaman kita dan akan mempermudah pemahaman tentang *interfacing* selanjutnya.
- Pemahaman yang diperlukan tersebut adalah ruang lingkup *interfacing* dan hubungan yang terkait dengannya.

1.1 PENGERTIAN INTERFACING

Interfacing (antar muka) adalah bagian dari disiplin ilmu komputer yang mempelajari teknik-teknik menghubungkan komputer dengan peralatan elektronika lainnya.

Sistem komputer yang berpusat pada pemroses utama (baik itu Mikroprosesor maupun Mikrokontroler) memiliki kemampuan yang besar dalam memecahkan masalah tetapi tidak ada manfaatnya tanpa menghubungkan dengan peralatan lainnya. Suatu teknik khusus diperlukan untuk dapat menghubungkan dengan peralatan-peralatan tersebut.

Menghubungkan pemroses utama dengan peralatan elektronik lainnya bukanlah persoalan yang mudah. Kita tidak dapat langsung menghubungkan pemroses utama dengan peralatan tersebut, disebabkan oleh hal-hal berikut

Terdapat beraneka ragam peralatan/ piranti yang memiliki metode operasi beragam.

Laju transfer data dalam piranti seringkali lebih lambat dibandingkan dengan laju transfer data dengan pemroses utama (Mikroprosesor). Piranti seringkali menggunakan format data yang berbeda dengan pemroses utama (Mikroprosesor).

Sebagai contoh :

Bandingkan antara Mikroprosesor dengan RS-232 (COMM)

	Mikroprosesor	RS-232C /Comm
Metode Operasi	Pararel	Serial
Transfer data	Sama dengan clock komputer (MBps)	20 Kbps
Format Data	Pararel (Bus)	Serial Asinkron

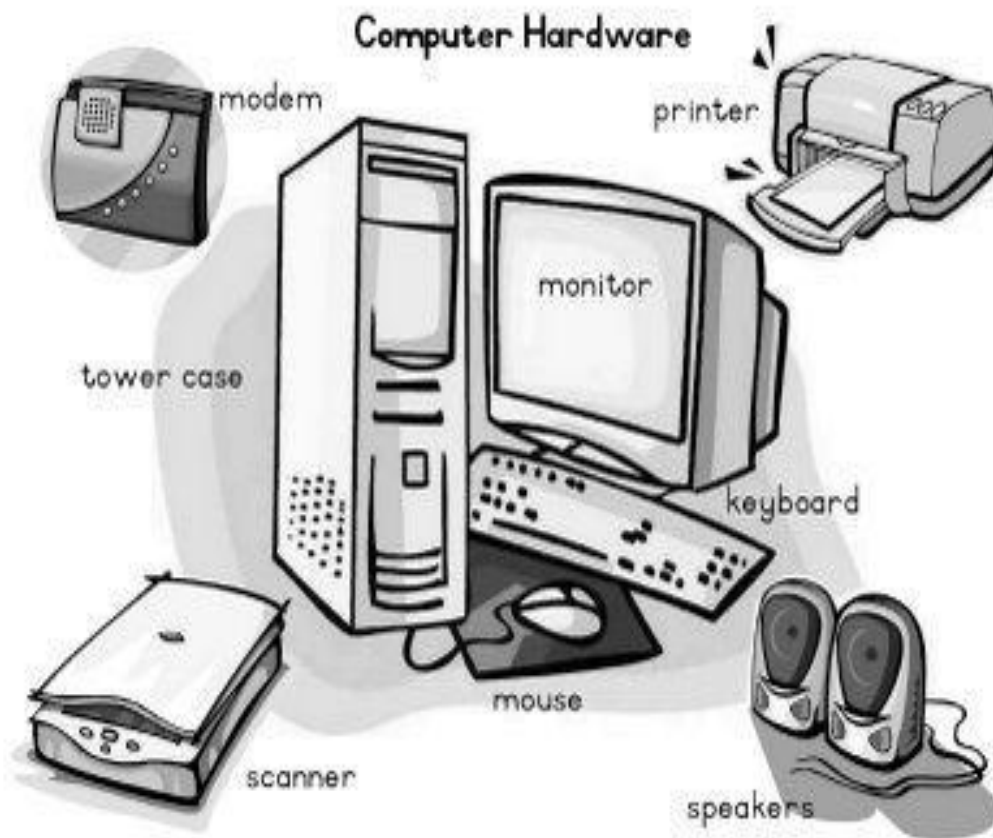
Jadi tidaklah praktis untuk menghubungkan mikroprosesor secara langsung dengan piranti yang ingin dijelankannya. Diperlukan suatu teknik untuk mem"perantara"kan pemroses utama (mikroprosesor) dengan dunia luar. Teknik ini dapat dijalankan melalui :

1. **Perangkat lunak**, berupa program, yakni suatu prosedur tertentu untuk menjalankan piranti. Dalam dunia komputer, program ini lebih dikenal sebagai *Driver/installer*. Adapula perangkat lunak yang dimasukkan kedalam perangkat keras yang disebut sebagai *Firmware*.
2. **Perangkat keras**, yakni berupa piranti khusus mulai dari serpih (IC) yang terintegrasi dalam sebuah papan induk (*Chipsets-Onboard*), berupa sebuah port atau bahkan terintegrasi kedalam papan yang ditancapkan pada system bus (*Card*).

1.2 RUANG LINGKUP INTERFACING

Interfacing bukanlah disiplin ilmu yang berdiri sendiri tetapi berkaitan erat dengan disiplin ilmu komputer lainnya. Pemahaman yang mendalam dari disiplin lainnya yang berkaitan dengan bahasan interfacing ini akan sangat membantu untuk memahami

materimateri yang disajikan selanjutnya. Disiplin ilmu komputer ini adalah Elektronika Analog dan Digital, Mikroprosesor, Organisasi dan Arsitektur Komputer, Komunikasi Data serta pendukung Bahasa Pemrograman, baik berbasis Teks seperti Bahasa Rakitan/ Assembly, C, Basic, Pascal maupun berbasis Grafis seperti Visual Basic, Visual C, Delphi bahkan berbasis Web seperti Java.



Gambar 1.1 peripheral komputer

Interface peripheral komputer?

Penghubung antar dua piranti (interface) atau penghubung antar perangkat-perangkat elektronik (komputer) perangkat - perangkat elektronik (komputer).

Interface = Pengantarmukaan= Pengantarmukaan

PeripheralPeripheral = Piranti / Perangkat= Piranti / Perangkat

Interface Peripheral KomputerInterface Peripheral Komputer

Penghubung antar komputer baik dengan komputer atauPenghubung antar komputer baik dengan komputer atau dengan perangkat lain (interfacing).

Interface peripheral komputer?

Penghubung antar dua piranti (interface) atau penghubung antarPenghubung antar dua piranti (interface) atau penghubung antar perangkat-perangkat elektronik (komputer).

Perangkat - perangkat elektronik (komputer).

Interkoneksi antar perangkat

PC to PC

PC to device

PC to μ Kontroler μ Processor to μ Kontroler

μ Processor to I/O μ Kontroler to I/O μ Processor to memory μ Kontroler to μ Kontroler

Struktur interfacing komputer terbagi 6Struktur interfacing komputer terbagi 6 layer (lapisan)layer (lapisan)

1.1. Electrical (physical)Electrical (physical)

2.2. SignalSignal

3.3. LogicLogic

4.4. ProtocolProtocol

5.5. CodeCode

6.6. AlgorithmicAlgorithm

Proses komunikasi data membutuhkan tiga jenis elemen dasar, yaitu :tiga jenis elemen dasar, yaitu :

- SUMBER DATASUMBER DATA
- MEDIA TRANSMISIMEDIA TRANSMISI
- PENERIMA DATAPENERIMA DATA

Proses komunikasi data pada jaringan disebut OSI (Open System Interconnection Open System Interconnection)

Apa itu Model OSI?

Model Open System Interconnection (OSI) menjelaskan tujuh lapisan yang digunakan sistem komputer untuk berkomunikasi melalui jaringan. Ini adalah model standar pertama untuk komunikasi jaringan, diadopsi oleh semua perusahaan komputer dan telekomunikasi besar pada awal 1980-an

Internet modern tidak didasarkan pada OSI, tetapi pada model TCP/IP yang lebih sederhana. Namun, model 7-layer OSI masih banyak digunakan, karena membantu memvisualisasikan dan mengkomunikasikan bagaimana jaringan beroperasi, dan membantu mengisolasi dan memecahkan masalah jaringan.

OSI diperkenalkan pada tahun 1983 oleh perwakilan dari perusahaan komputer dan telekomunikasi besar, dan diadopsi oleh ISO sebagai standar internasional pada tahun 1984. layer model osi terbagi layer model osi

terbagi upper layer upper layer lower layer lower

layer



Gambar 1.2 .Open System Interconnection (OSI) layer

Kami akan menjelaskan lapisan OSI "atas ke bawah" dari lapisan aplikasi yang secara langsung melayani pengguna akhir, hingga ke lapisan fisik.

7. Lapisan Aplikasi

Lapisan aplikasi digunakan oleh perangkat lunak pengguna akhir seperti browser web dan klien email. Ini menyediakan protokol yang memungkinkan perangkat lunak untuk mengirim dan menerima informasi dan menyajikan data yang berarti kepada pengguna. Beberapa contoh protokol lapisan aplikasi adalah Hypertext Transfer Protocol (HTTP),

File Transfer Protocol (FTP), Post Office Protocol (POP), Simple Mail Transfer Protocol (SMTP), dan Domain Name System (DNS).

6. Lapisan Presentasi

Lapisan presentasi menyiapkan data untuk lapisan aplikasi. Ini mendefinisikan bagaimana dua perangkat harus menyandikan, mengenkripsi, dan mengompresi data sehingga diterima dengan benar di ujung yang lain. Lapisan presentasi mengambil data yang dikirimkan oleh lapisan aplikasi dan mempersiapkannya untuk transmisi melalui lapisan sesi.

5. Lapisan Sesi

Lapisan sesi membuat saluran komunikasi, yang disebut sesi, antar perangkat. Ini bertanggung jawab untuk membuka sesi, memastikan sesi tetap terbuka dan berfungsi saat data sedang ditransfer, dan menutupnya saat komunikasi berakhir. Lapisan sesi juga dapat mengatur pos pemeriksaan selama transfer data—jika sesi terganggu, perangkat dapat melanjutkan transfer data dari pos pemeriksaan terakhir.

4. Lapisan Transportasi

Lapisan transport mengambil data yang ditransfer di lapisan sesi dan memecahnya menjadi "segmen" di ujung transmisi. Ini bertanggung jawab untuk merakit kembali segmen di ujung penerima, mengubahnya kembali menjadi data yang dapat digunakan oleh lapisan sesi. Lapisan transport melakukan kontrol aliran, mengirimkan data dengan kecepatan yang sesuai dengan kecepatan koneksi perangkat penerima, dan kontrol kesalahan, memeriksa apakah data yang diterima salah dan jika tidak, memintanya lagi.

3. Lapisan Jaringan

Lapisan jaringan memiliki dua fungsi utama. Salah satunya adalah memecah segmen menjadi paket jaringan, dan memasang kembali paket di sisi penerima. Yang lainnya adalah merutekan paket dengan menemukan jalur terbaik melintasi jaringan fisik. Lapisan jaringan menggunakan alamat jaringan (biasanya alamat Protokol Internet) untuk merutekan paket ke node tujuan.

2. Lapisan Tautan Data

Lapisan data link membuat dan mengakhiri koneksi antara dua node yang terhubung secara fisik pada jaringan. Ini memecah paket menjadi bingkai dan mengirimkannya dari

sumber ke tujuan. Lapisan ini terdiri dari dua bagian—*Logical Link Control (LLC)*, yang mengidentifikasi protokol jaringan, melakukan pemeriksaan kesalahan dan menyinkronkan frame, dan *Media Access Control (MAC)* yang menggunakan alamat MAC untuk menghubungkan perangkat dan menentukan izin untuk mengirim dan menerima data.

1. Lapisan Fisik

Lapisan fisik bertanggung jawab atas kabel fisik atau koneksi nirkabel antara node jaringan. Ini mendefinisikan konektor, kabel listrik atau teknologi nirkabel yang menghubungkan perangkat, dan bertanggung jawab untuk transmisi data mentah, yang hanya serangkaian 0 dan 1, sambil menjaga kontrol *bit rate*.

BAB II

CPU dan Mikrokontroler

Mata Kuliah : TEKNIK INTERFACE

Capaian Pembelajaran :

Mahasiswa Mampu menerapkan dan mengembangkan teknik antarmuka, dalam sistem mikrokomputer sesuai dengan kebutuhan desain yang harus dipenuhi Kemampuan

Akhir yang Diharapkan :

Mahasiswa Memahami tentang CPU dan Mikrokomputer

Alokasi Waktu : 3 X 50 menit

Pertemuan ke : 2 Indikator :

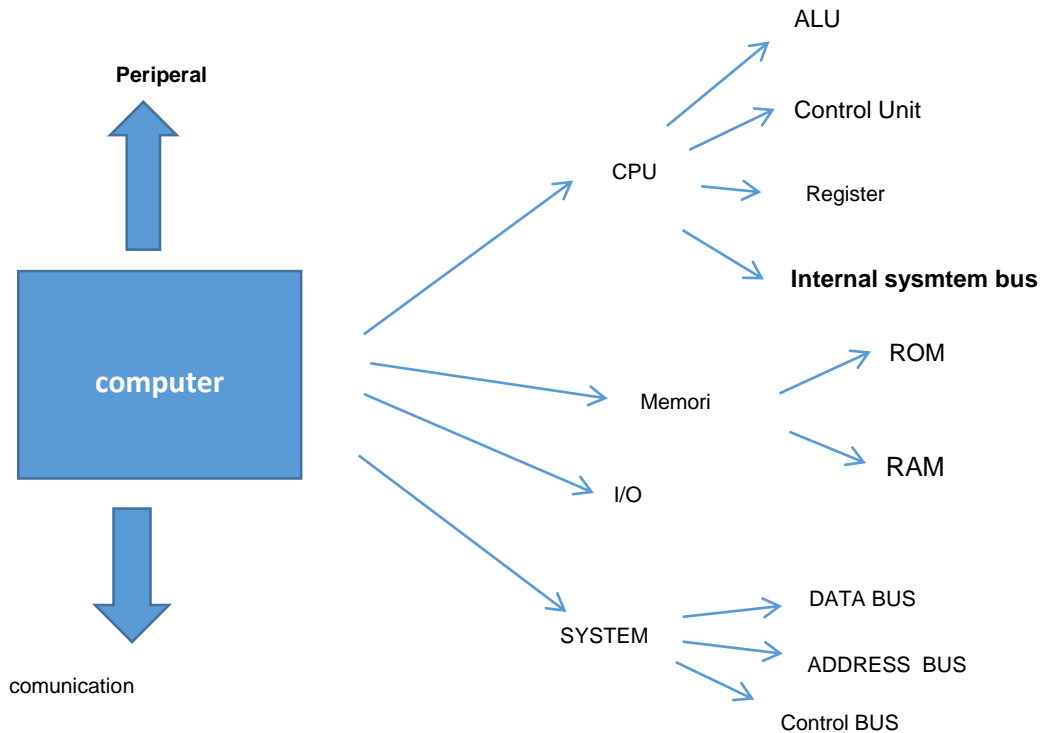
1. Mahasiswa akan mampu menjelaskan antarmuka yang disediakan oleh mikrokontroler AT89S52 (MCS51)
2. Mahasiswa akan mampu menjelaskan antarmuka yang disediakan oleh mikrokontroler AVR ATmega32P
3. Mahasiswa akan mampu mengaplikasikan rangkaian sumber daya yang stabil ke system mikrokomputer
4. Mahasiswa akan mampu mengaplikasikan rangkaian clock ke sistem mikrokomputer sesuai dengan kebutuhan dan konstrain kecepatan mikrokomputer
5. Mahasiswa akan mampu mengaplikasikan rangkaian reset ke system mikrokomputer dengan

BAB 2

CPU dan Mikrokontroler

Mikroprosesor adalah chip elektronik yang dapat diprogram yang memiliki komputasi dan keputusan membuat kemampuan yang mirip dengan unit pemrosesan pusat komputer. Setiap sistem berbasis mikroprosesor yang memiliki jumlah sumber daya terbatas disebut mikrokomputer. Pengantar Mikroprosesor

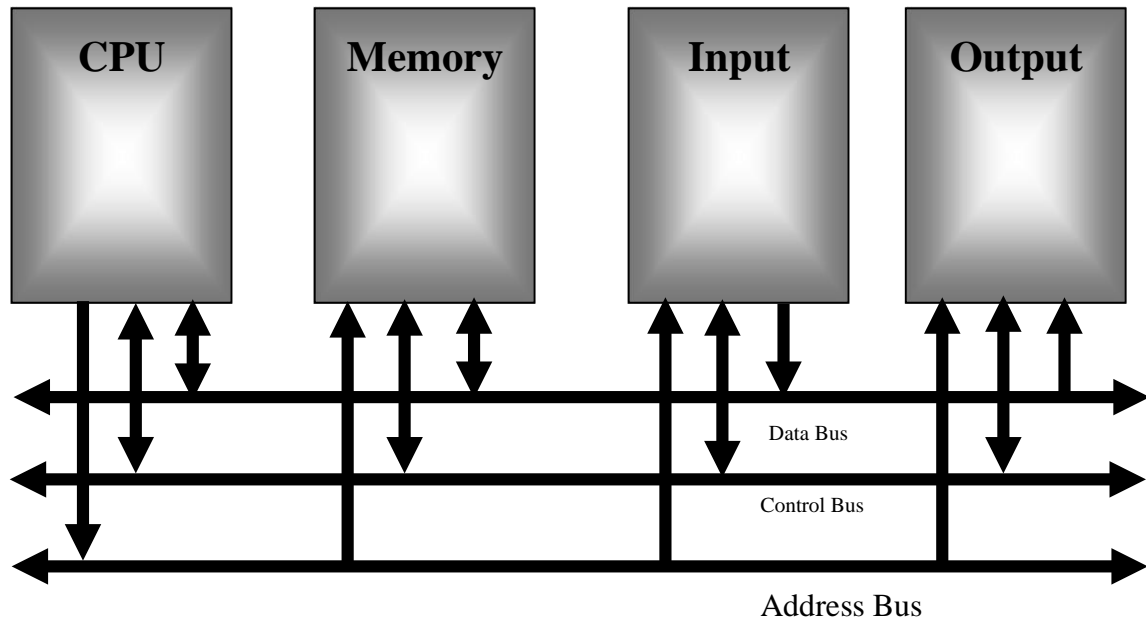
Mikroprosesor adalah prosesor komputer yang menggabungkan fungsi unit pemrosesan pusat (CPU) komputer pada sirkuit terpadu tunggal (IC), atau paling banyak beberapa sirkuit terpadu Mikroprosesor adalah multiguna, clock-driven, register based, sirkuit terintegrasi digital yang menerima data biner sebagai input, memprosesnya sesuai dengan instruksi yang tersimpan dalam memorinya, dan memberikan hasil sebagai keluaran. Mikroprosesor mengandung logika kombinasional dan sekuensial logika digital. Mikroprosesor beroperasi pada angka dan simbol yang diwakili dalam sistem bilangan biner.



Gb. 2.1. bagian - bagian system komputer

Struktur model computer menurut Von Neumann Gambar 2.2. terdiri dari

- Input
- Output
- Memory
- CPU Central Processing Unit



Gambar 2.2. Model arsitektur komputer Von Neumann

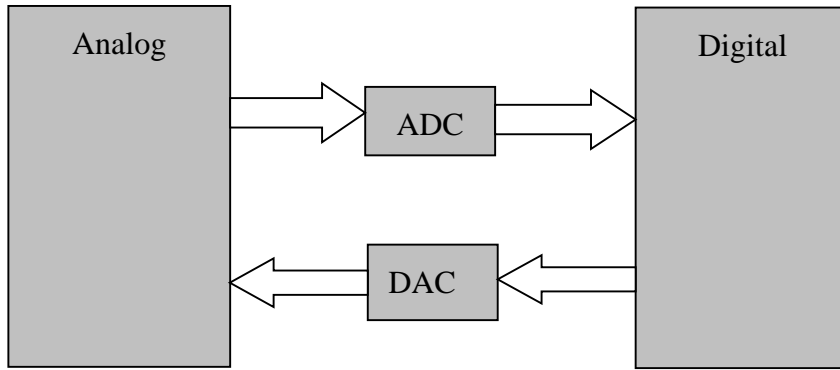
Perangkat Keras, Perangkat Lunak, dan Firmware

- Perangkat keras adalah nama yang diberikan untuk perangkat fisik dan sirkuit komputer.
- Perangkat lunak mengacu pada program yang ditulis untuk komputer.
- Firmware adalah istilah yang diberikan untuk program yang disimpan dalam ROM atau di perangkat yang dapat diprogram yang secara permanen menyimpan informasi yang tersimpan

Kategori Antarmuka Periferal

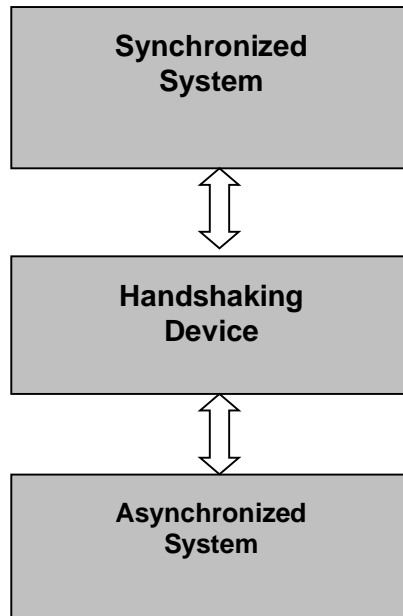
- Antarmuka Analog
- Antarmuka Digital

Untuk menghubungkan dua periferal yang berbeda system salah satunya digital dan yang lainnya analog kita harus menambahkan analog to digital converter (ADC) dan digital to analog converter (DAC) (Gambar 2.3)



Gambar 2.3. Antarmuka yang Disinkronkan dan Tidak Disinkronkan

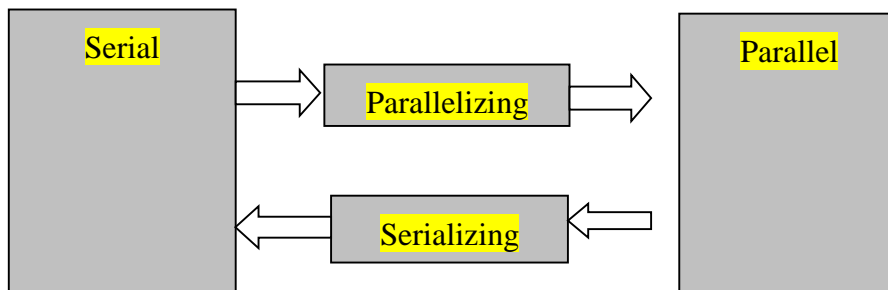
- Dua kategori penting dari antarmuka digunakan untuk menghubungkan periferil: yang pertama adalah antarmuka yang disinkronkan yang bergantung pada jam untuk mengatur transfer data. Yang kedua adalah antarmuka asinkron yang dapat dicapai tanpa jam. Untuk menghubungkan periferil ini bersama-sama, kita memerlukan adaptor handshaking yang mengatur pertukaran data di antara komponen (Gambar 2.4)..



Gambar 2.4. Proses *interface* . Synchronisasi dan Asynchron

Antar muka / in terface Serial dan Paralel

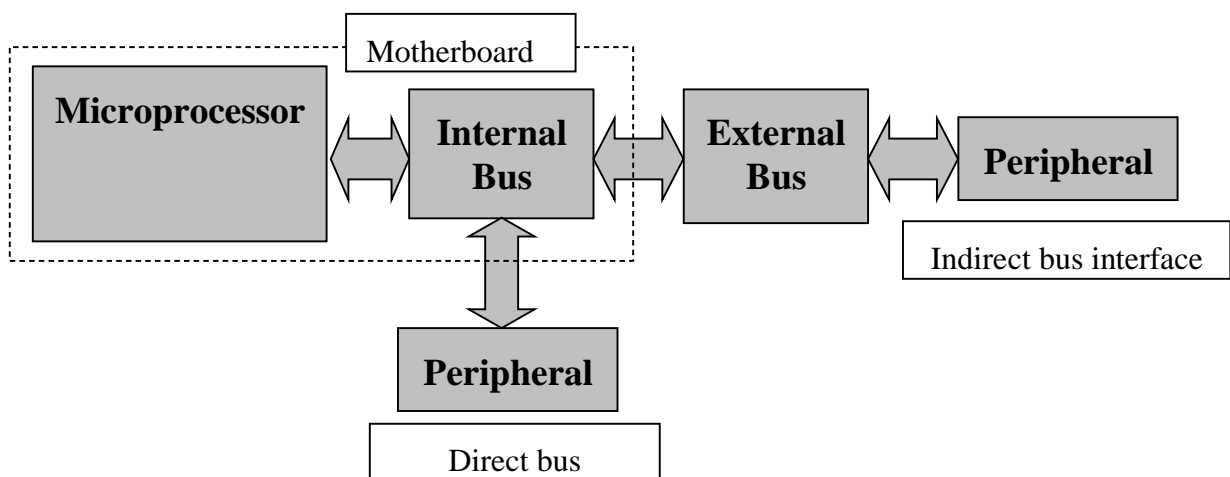
- Untuk menghubungkan dua periferal salah satunya adalah paralel dan yang kedua adalah serial kita harus menggunakan tahap paralelisasi dan serialisasi untuk menghubungkan keduanya. Tahap paralelisasi mengubah pulsa serial menjadi data paralel sedangkan tahap serialisasi mengubah data paralel menjadi pulsa serial. (Gambar 2.5)



Gambar 2.5 . Interface paralel serial Antar

muka pada Microprocessor

- Mikroprosesor dapat dihubungkan dengan periferal melalui beberapa teknik, yaitu:
 - Antarmuka ruang memori
 - Antarmuka port I/O seperti port serial dan paralel
 - Antarmuka bus langsung/internal menggunakan bus internal seperti ISA, EISA, PCI, AGP, USB, SATA, IDE, SCSI, ...
 - Antarmuka bus tidak langsung/eksternal menggunakan bus eksternal seperti GPIB, CAMAC, CAN, dll..

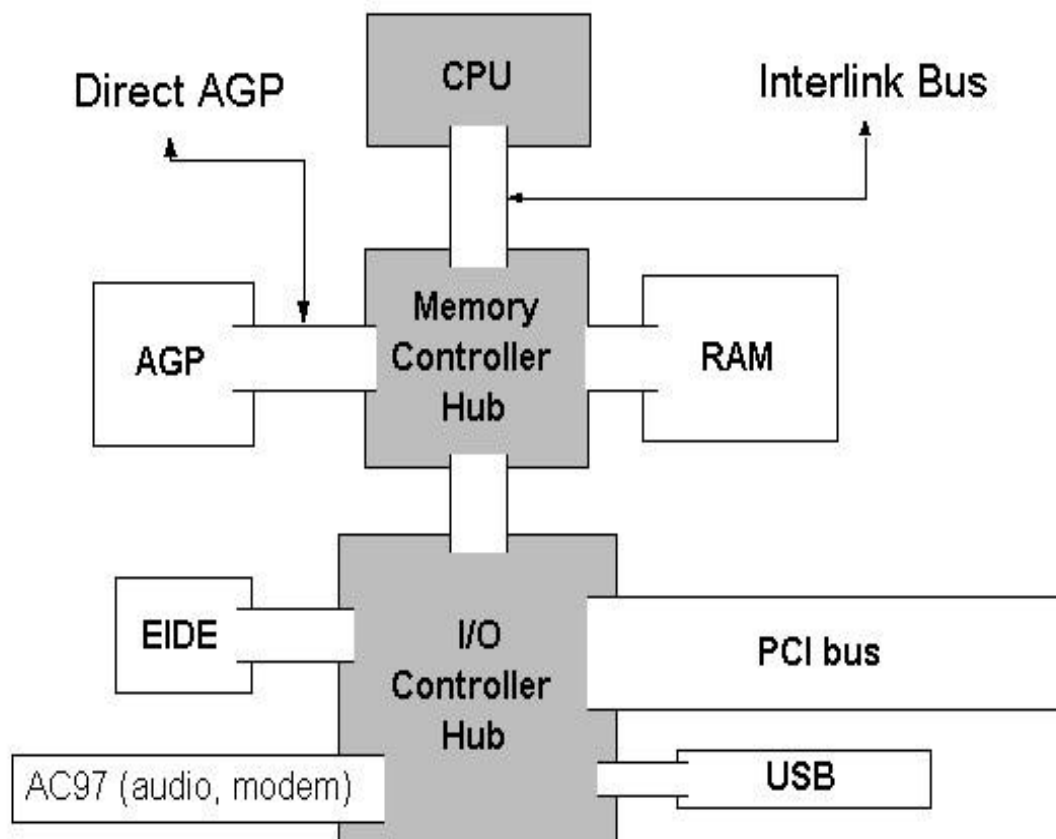


Gambar 2.6. Bus Interface Direct dan Indirect

Arsitektur USB 2.0

Desain Arsitektur USB 2.0 mempunyai topologi yang asimetris, yang terdiri dari sebuah host, memiliki banyak port USB downstream, dan beberapa perangkat periperal yang terhubung dalam deretan topologi star. USB hub tambahan dapat dimasukkan dalam deretan topologi tersebut, yang memungkinkan membentuk percabangan dalam sebuah struktur tree hingga mencapai lima tingkat deretan. Sebuah host USB dapat mengimplementasikan beberapa host controller, dan masing-masing host controller dapat memberikan servis kepada satu atau lebih port USB. Host controller tersebut mampu memberikan servis hingga mencapai lebih dari 127 perangkat, termasuk perangkat hub jika ada.

Perangkat USB terhubung dalam seri melalui hub. Sebuah hub dikenal sebagai root hub yang dibentuk ke dalam host controller. Sebuah perangkat physical USB dapat terdiri dari beberapa logical sub-device yang disebut sebagai device functions. Sebuah perangkat tunggal dapat menyediakan beberapa fungsi, misalnya, webcam (fungsi perangkat video) dengan built-in microphone (fungsi perangkat audio). Jenis perangkat semacam ini disebut sebagai perangkat komposit. Sebuah alternatif untuk hal ini adalah dengan menggabungkan perangkat di mana setiap logical device diberikan sebuah alamat yang berbeda oleh host kemudian dihubungkan ke built-in hub yang terhubung pada kawat USB secara fisik.



Gambar 2.7 Arsitektur system Hub

Klasifikasi Sistem Bus

Sebuah sistem komputer terdiri dari tiga komponen utama: prosesor, memori, unit, dan subsistem input/output (I/O). Jaringan interkoneksi memfasilitasi komunikasi di antara ketiga komponen ini, seperti yang ditunjukkan pada Gambar 21 jaringan interkoneksi disebut sistem bus.

Klasifikasi Sistem Bus Menurut Fungsi

- Bus Data
- Bus Kontrol
- Alamat Bus

Menurut Lokasi

- ISA Internal / Langsung, EISA, PCI, AGP, USB, SCSI, IDE, SATA, ...

- GPIB Eksternal / Tidak Langsung, CAMAC, CAN, ...

Sesuai Tujuan

- Bus Tujuan Umum
- Bus Tujuan Khusus

Bus tidak hanya koneksi kabel tetapi juga perangkat keras, arsitektur bus, protokol, perangkat lunak, dan pengontrol bus

Definisi Penting

PnP

PnP: Singkatan dari Plug and Play,

PnP adalah kemampuan komputer untuk mendeteksi dan mengkonfigurasi perangkat keras baru secara otomatis, tanpa persyaratan pengguna untuk mengkonfigurasi perangkat keras secara fisik dengan jumper atau dipswitch. Agar Plug and Play dapat beroperasi dengan benar pada komputer yang kompatibel dengan IBM, pengguna harus memiliki yang berikut ini:

- BIOS mendukung Plug and Play
- Sistem operasi yang mendukung PnP
- Periferal dengan dukungan PnP.

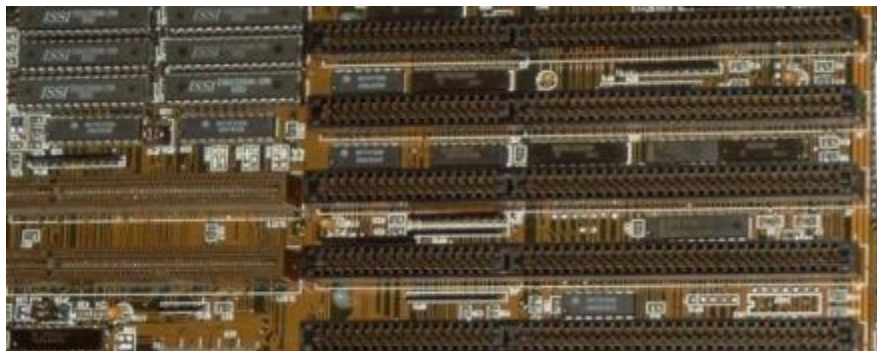
Saat ini semua komputer baru memiliki kemampuan PnP. Komputer yang menjalankan Microsoft Windows tidak lagi mendukung perangkat non PnP.

Throughput (Baud-rate, Kecepatan): Juga dikenal sebagai "kecepatan komunikasi", throughput adalah nilai numerik yang digunakan untuk menggambarkan jumlah total data yang ditransfer yang ditransfer melalui komputer atau perangkat pada waktu tertentu. Angka ini biasanya direpresentasikan dalam bit per second (bps) atau byte per second (Bps).

Hak milik Istilah yang digunakan untuk menggambarkan produk yang hanya kompatibel dengan jenis perangkat keras, perangkat lunak, komputer, atau pabrikan tertentu. Saat mengacu pada perangkat keras komputer, Anda disarankan untuk tidak memilih perangkat berpemilik karena dapat mengurangi kompatibilitas dan umumnya kemampuan untuk meningkatkan produk tersebut di masa mendatang.

ISA BUS

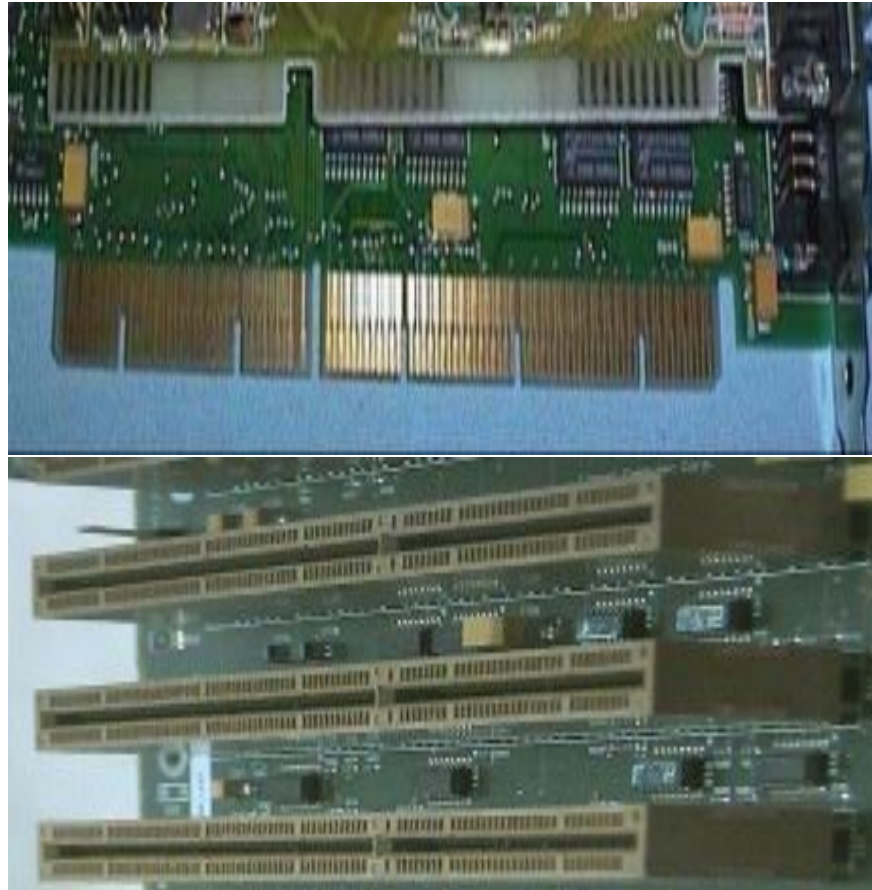
Diperkenalkan oleh IBM, ISA atau Arsitektur Standar Industri (Gambar 2.8) pada awalnya merupakan bus 8-bit yang kemudian diperluas menjadi bus 16-bit pada tahun 1984. Ketika BUS ini awalnya dirilis, itu adalah BUS berpemilik, yang hanya mengizinkan IBM untuk membuat periferal dan antarmuka yang sebenarnya. Namun pada awal 1980-an produsen lain menciptakan bus.



Gambar 2.8. ISA BUS

The EISA Bus

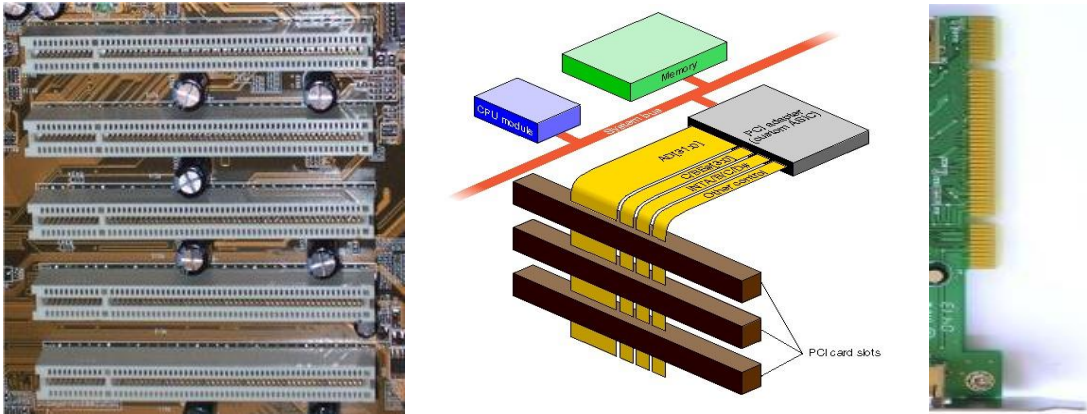
Singkatan dari Extended Industry Standard Architecture, EISA diumumkan September 1988. EISA adalah bus komputer yang dirancang oleh 9 pesaing untuk bersaing dengan MCA BUS IBM. Pesaing tersebut adalah AST Research, Compaq, Epson, Hewlett Packard, NEC, Olivetti, Tandy, WYSE, dan Zenith Data Systems. Bus EISA menyediakan slot 32-bit pada kecepatan siklus 8,33 MHz untuk digunakan dengan 386DX, atau prosesor yang lebih tinggi. Selain itu EISA dapat menampung kartu ISA 16bit di baris pertama. Sayangnya, sementara bus EISA kompatibel ke belakang dan bukan merupakan bus berpemilik, bus EISA tidak pernah digunakan secara luas dan tidak lagi ditemukan di komputer saat ini.



Gambar 2.8. EISA Bus

PCI BUS

Diperkenalkan oleh Intel pada tahun 1992, direvisi pada tahun 1993 menjadi versi 2.0, dan kemudian direvisi pada tahun 1995 menjadi PCI 2.1. PCI adalah kependekan dari Peripheral Component Interconnect dan merupakan bus komputer 32-bit yang juga tersedia sebagai bus 64-bit saat ini. Bus PCI adalah bus yang paling umum digunakan dan ditemukan di komputer saat ini.



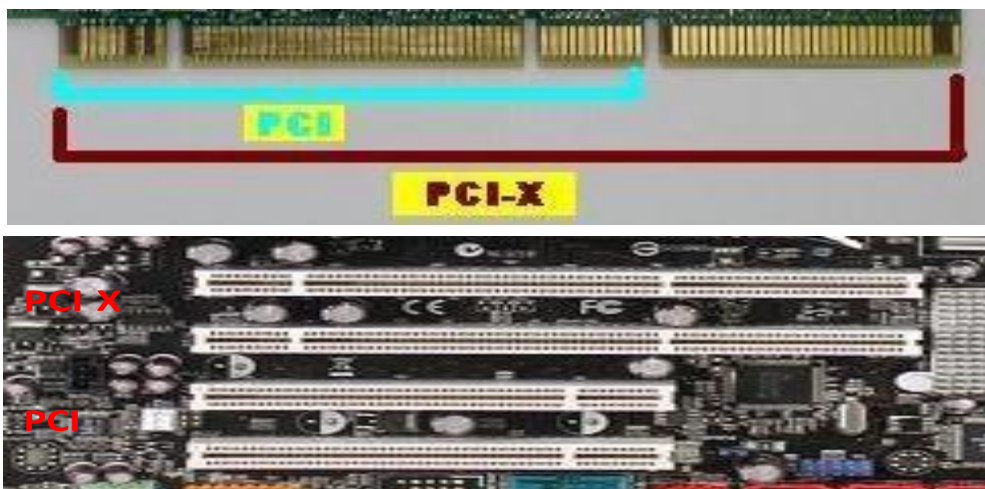
Gambar 2.9. PCI BUS

- PCI-X adalah bus performa tinggi yang dirancang untuk memenuhi tuntutan teknologi I/O yang meningkat seperti Fibre Channel, Gigabit Ethernet, dan Ultra3 SCSI. Kemampuan

PCI-X meliputi: • Kecepatan bus hingga 133 MHz, bandwidth 64-Bit, throughput 1GB/dtk

- Pengoperasian bus yang lebih efisien untuk antarmuka yang lebih mudah • Transaksi Terpisah memungkinkan perangkat indikator untuk membuat hanya satu permintaan data dan melepaskan bus. Alih-alih terus-menerus perlu polling bus untuk tanggapan. • Byte Count yang memungkinkan indikator untuk menentukan terlebih dahulu jumlah byte tertentu yang diminta, menghilangkan inefisiensi prefetches spekulatif.

Kompatibilitas mundur



Gambar 2.8. PCI-X Bus

PCI-X

PCI-X, kependekan dari Peripheral Component Interconnect eXtended, adalah bus komputer dan standar kartu ekspansi yang meningkatkan bus lokal PCI 32-bit untuk bandwidth yang lebih tinggi yang sebagian besar diminta oleh server dan workstation. Ini menggunakan protokol yang dimodifikasi untuk mendukung kecepatan clock yang lebih tinggi (hingga 133 MHz), tetapi sebaliknya serupa dalam implementasi listrik. PCI-X 2.0 menambahkan kecepatan hingga 533 MHz,[2]:23 dengan pengurangan level sinyal listrik.

Slot tersebut secara fisik adalah slot PCI 3,3 V, dengan ukuran, lokasi, dan penetapan pin yang sama persis. Spesifikasi kelistrikannya kompatibel, tetapi lebih ketat. Namun, meskipun sebagian besar slot PCI konvensional adalah versi 32-bit dengan panjang 85 mm, sebagian besar perangkat PCI-X menggunakan slot 64-bit dengan panjang 130 mm, sehingga konektor PCI 64-bit dan dukungan PCI-X dianggap sama. .

PCI-X sebenarnya sepenuhnya ditentukan untuk konektor PCI 32- dan 64-bit dan PCI-X 2.0 menambahkan varian 16-bit untuk aplikasi tertanam.

Ini telah digantikan dalam desain modern oleh PCI Express yang terdengar serupa (secara resmi disingkat PCIe), dengan konektor fisik yang sama sekali berbeda dan desain listrik yang sangat berbeda, memiliki satu atau lebih jalur koneksi serial yang sempit tetapi cepat alih-alih jumlah koneksi yang lebih lambat secara paralel.

Berbagai slot PCI. Dari atas ke bawah:

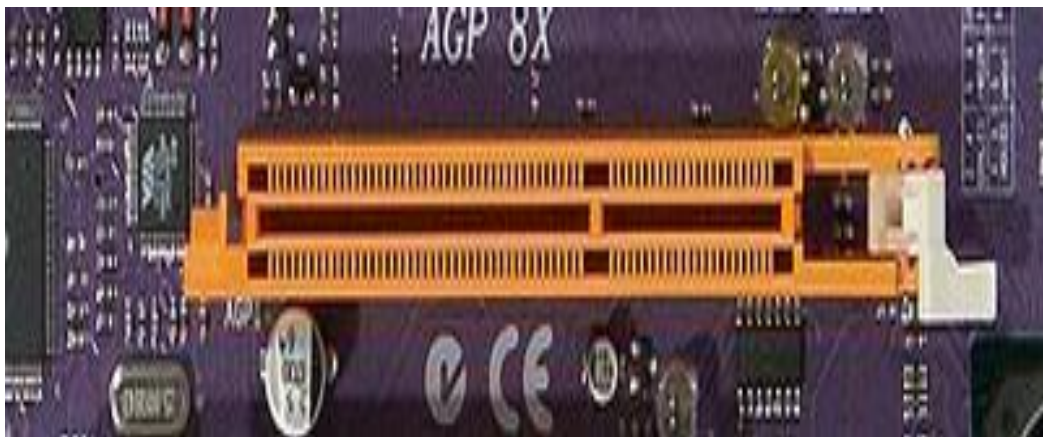
- PCI Express ×4
- PCI Express ×16
- PCI Express ×1
- PCI Express ×16 PCI konvensional (32-bit)



Gambar 2.9. PCI-Slots

Bus: AGP

AGP: Diperkenalkan oleh Intel pada tahun 1997, AGP atau Advanced Graphic Port adalah bus 32-bit yang dirancang untuk kebutuhan grafis 3-D yang tinggi. AGP memiliki saluran langsung ke memori komputer yang memungkinkan elemen 3-D disimpan dalam memori sistem alih-alih memori video. Agar AGP dapat bekerja di komputer harus memiliki slot AGP yang disertakan dengan sebagian besar mesin Pentium II dan Pentium III. Komputer juga harus menjalankan Windows 95 OSR2.1, Windows 98, Windows 98 SE, Windows 2000, Windows ME atau lebih tinggi.



Gambar 2.10 AGP atau Advanced Graphic Port

Accelerated Graphics Port, sering disingkat AGP, adalah jenis koneksi standar untuk kartu video internal. Umumnya, Port Grafis Dipercepat mengacu pada slot ekspansi

aktual pada motherboard yang menerima kartu video AGP serta jenis kartu video itu sendiri.

Perbedaan Antara AGP dan PCI

AGP diperkenalkan oleh Intel pada tahun 1997 sebagai pengganti antarmuka Peripheral Component Interconnect (PCI) yang lebih lambat. AGP menyediakan jalur komunikasi langsung ke CPU dan RAM, yang pada gilirannya memungkinkan rendering grafis yang lebih cepat.

Salah satu peningkatan besar yang dimiliki AGP dibandingkan antarmuka PCI adalah cara kerjanya dengan RAM. Disebut memori AGP, atau memori non-lokal, AGP dapat mengakses memori sistem secara langsung daripada hanya mengandalkan memori kartu video.

Memori AGP memungkinkan kartu AGP untuk menghindari keharusan menyimpan peta tekstur (yang mungkin menggunakan banyak memori) pada kartu itu sendiri karena kartu tersebut menyimpannya di memori sistem. Ini berarti tidak hanya kecepatan keseluruhan AGP yang ditingkatkan dibandingkan PCI, tetapi juga batas ukuran unit tekstur tidak lagi ditentukan oleh jumlah memori di kartu grafis.

Kartu grafis PCI menerima informasi dalam "kelompok" sebelum dapat menggunakannya, bukan sekaligus. Misalnya, sementara kartu grafis PCI akan mengumpulkan tinggi, panjang, dan lebar gambar pada tiga waktu yang berbeda, dan kemudian menggabungkannya bersama untuk membentuk sebuah gambar, AGP bisa mendapatkan semua informasi itu secara bersamaan. Hal ini membuat grafik lebih cepat dan lebih halus daripada yang Anda lihat dengan kartu PCI.

Sebuah bus PCI biasanya berjalan pada kecepatan 33 MHz, memungkinkan untuk mentransfer data pada 132 MB/s. Menggunakan tabel di atas, Anda dapat melihat bahwa AGP 3.0 mampu berjalan lebih dari 16 kali kecepatan untuk mentransfer data lebih cepat, dan bahkan AGP 1.0 melebihi kecepatan PCI dengan dua faktor.

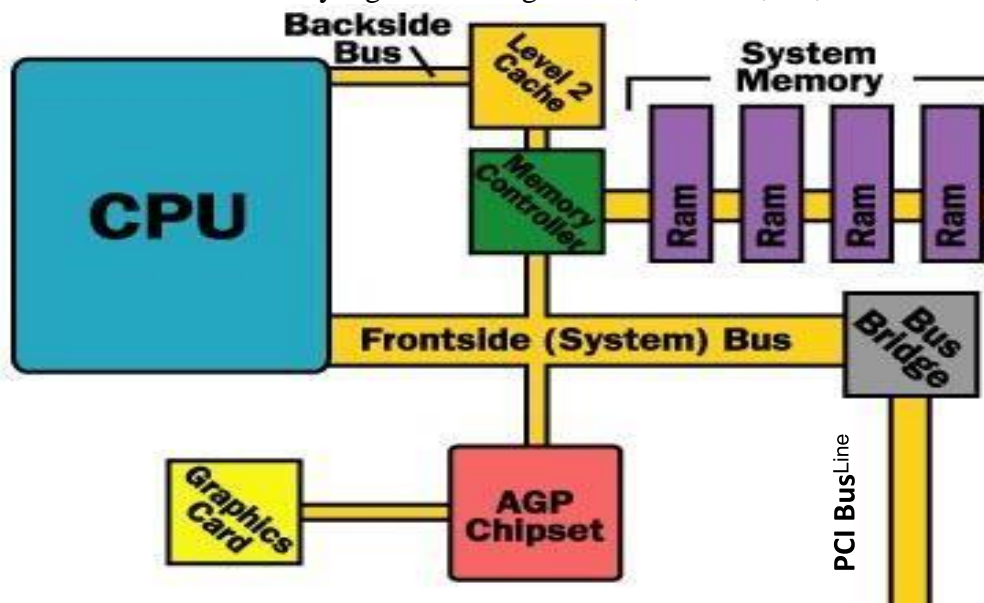
Sementara AGP menggantikan PCI untuk grafis, PCIe (PCI Express) telah menggantikan AGP sebagai antarmuka kartu video standar, yang hampir sepenuhnya menggantikannya pada tahun 2010.

Kompatibilitas AGP

Motherboard yang mendukung AGP akan memiliki slot yang tersedia untuk kartu video AGP atau akan memiliki AGP onboard.

Kartu video AGP 3.0 dapat digunakan pada motherboard yang hanya mendukung AGP 2.0, tetapi akan dibatasi pada apa yang didukung oleh motherboard, bukan pada apa yang didukung oleh kartu grafis. Dengan kata lain, motherboard tidak akan mengizinkan kartu video untuk bekerja lebih baik hanya karena itu adalah kartu AGP 3.0; motherboard itu sendiri tidak mampu kecepatan seperti itu (dalam skenario ini).

Beberapa motherboard yang hanya menggunakan AGP 3.0 mungkin tidak mendukung kartu AGP 2.0 yang lebih lama. Jadi, dalam skenario terbalik dari atas, kartu video bahkan mungkin tidak berfungsi kecuali jika mampu bekerja dengan antarmuka yang lebih baru. Tersedia slot AGP universal yang mendukung kartu 1,5 V dan 3,3 V, serta kartu universal.



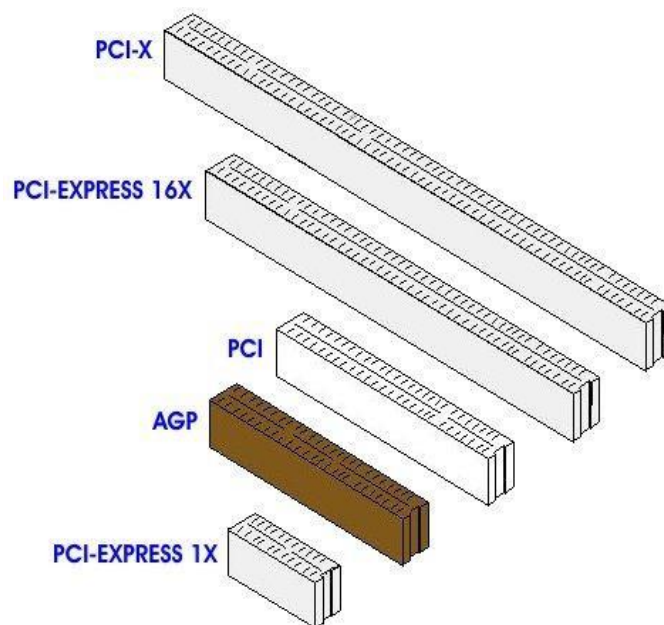
Gambar arsitektur The Advanced Graphic Port, AGP

Motherboard Slots

Motherboard adalah papan sirkuit elektronik di komputer yang saling terhubung perangkat keras yang terpasang padanya, artinya, semua perangkat keras sistem. minimal mencakup satu atau lebih Centra Processing Unit (CPU), dan yang utama aktivitas pemrosesan komputer berlangsung di dalamnya. Namun, yang lain terhubung papan sirkuit tercetak mungkin berisi pra-pemrosesan atau pasca-pemrosesan mereka sendiri CPU, untuk mengambil sebagian beban dari motherboard; ini, bersama dengan yang lain.

Slot

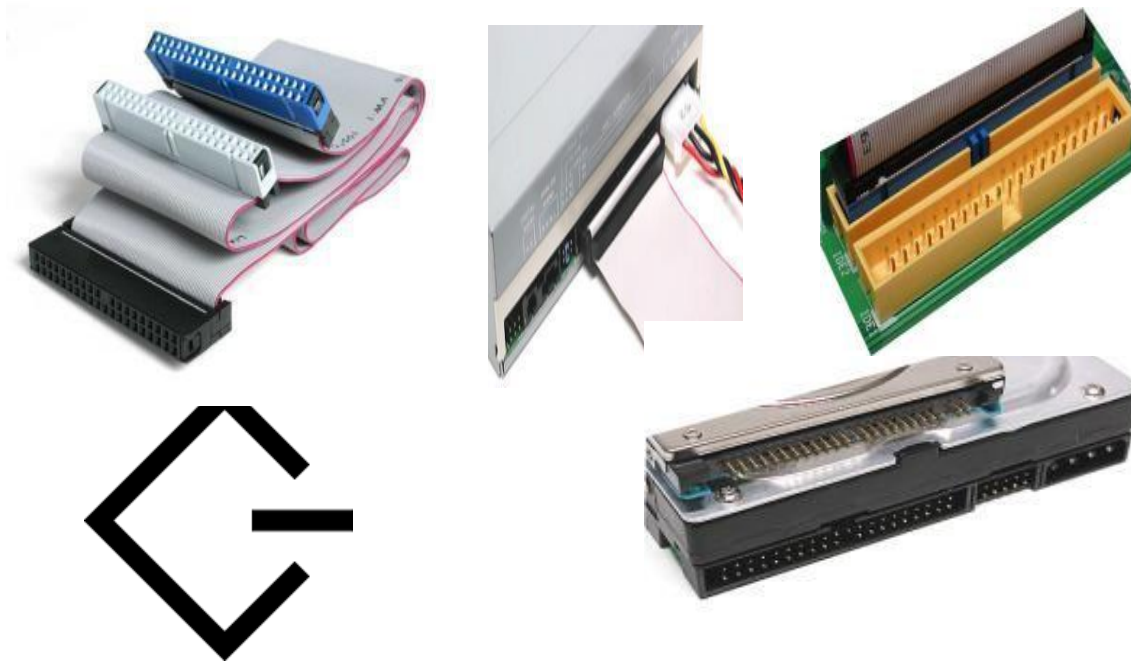
Pada PC desktop biasa, sejumlah slot disediakan pada motherboard untuk memasang perangkat gambar 2.11. Salah satu penggunaan slot ini adalah untuk memperluas atau menambah port yang tersedia untuk perangkat eksternal, seperti yang disebutkan di atas. Grafik, audio, dan jaringan dapat digabungkan pada motherboard, tetapi sering ditangani oleh papan sirkuit (anak) yang ditambahkan ke dalam slot. Untuk memungkinkan ekspansi melalui perangkat eksternal, laptop sering kali memiliki slot (atau port) eksternal.



Gambar 2.11. Ekspansion **Slot**

Buses: IDE - Parallel ATA

Ini adalah bus paralel untuk transfer data dari hard disk drive, CD-ROM, drive Tape Backup, drive Zip, DVD-ROM, atau perangkat ATA apa pun ke komputer berkemampuan IDE. Juga dikenal sebagai Parallel ATA, kontras dengan Serial ATA. Versi terbaru dilengkapi kabel pita 40 pin, 80 kawat untuk menghubungkan motherboard ke drive. Setiap kabel tersebut dapat mendukung hingga maksimal dua perangkat, dengan satu drive pada kabel yang dikonfigurasi sebagai drive master, dan yang lainnya sebagai slave. Pengaturan ini biasanya ditangani oleh blok jumper kecil di suatu tempat di drive.

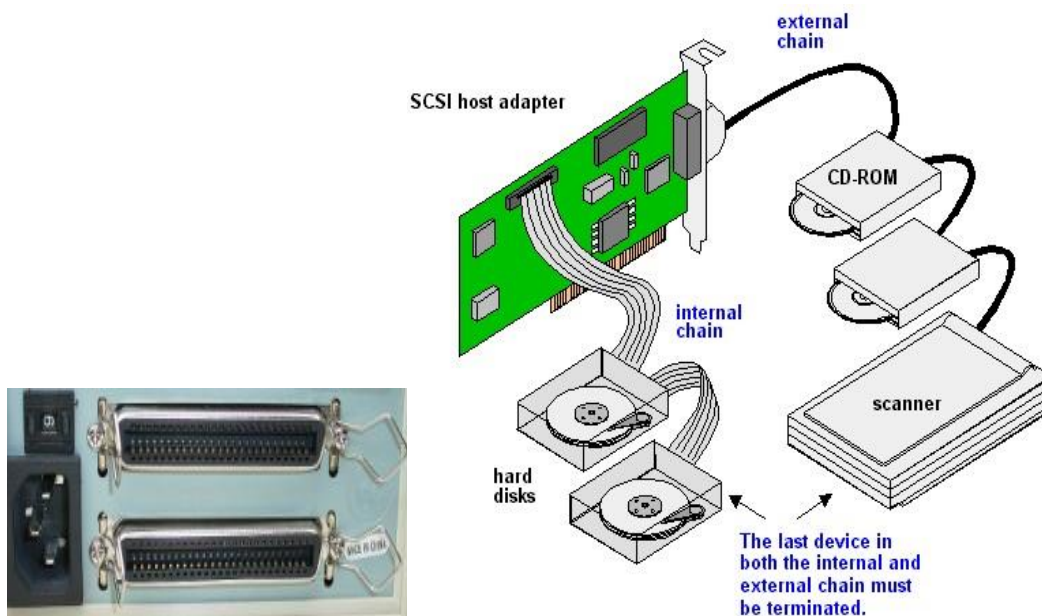


Gambar 2.12. **Buses: IDE - Parallel ATA**

Buses Small Computer System Interface, or SCSI

Antarmuka Sistem Komputer Kecil, atau SCSI (diucapkan scuzzy), adalah seperangkat standar untuk menghubungkan dan mentransfer data secara fisik antara komputer dan perangkat perifer. Standar SCSI mendefinisikan perintah, protokol, dan antarmuka listrik dan optik. SCSI paling sering digunakan untuk hard disk dan tape drive, tetapi dapat menghubungkan berbagai perangkat lain, termasuk pemindai dan drive CD. Standar SCSI mendefinisikan set perintah untuk jenis perangkat perifer tertentu; kehadiran "tidak diketahui" sebagai salah satu dari jenis ini berarti bahwa secara teori dapat digunakan sebagai antarmuka ke hampir semua perangkat, tetapi standarnya sangat

pragmatis dan ditujukan untuk persyaratan komersial. SCSI adalah antarmuka yang cerdas, periferal, buffered, peer to peer. Ini menyembunyikan kompleksitas format fisik. Setiap perangkat terhubung ke bus SCSI dengan cara yang sama. Hingga 8 atau 16 perangkat dapat dilampirkan ke satu bus. Mungkin ada sejumlah host dan perangkat periferal tetapi harus ada setidaknya satu host. SCSI menggunakan sinyal jabat tangan antar perangkat, SCSI-1, SCSI-2 memiliki opsi pemeriksaan kesalahan paritas. Dimulai dengan SCSI-U160 (bagian dari SCSI-3) semua perintah dan data diperiksa kesalahannya oleh checksum CRC32.



Gambar 2.13. Buses SCSI

Buses: Serial ATA, SATA

Serial ATA, atau **bus komputer SATA**, adalah sebuah antarmuka penyimpanan kepada menghubungkan *host bus adapter* kepada perangkat penyimpanan massal seperti *hard disk drive* dan *optical drive*. Mulai pada tahun 2004, *SATA host adapter* terintegrasi ke hampir semua motherboard modern, elok itu desktop maupun laptop. Serial ATA ini didesain kepada menggantikan standar ATA (AT Attachment) yang lebih tua (juga dikenal sebagai EIDE). SATA dapat menggunakan perintah-perintah tingkat rendah yang sama, namun *SATA host-adapter* dan perangkat yang terkoneksi mengadakan komunikasi melintasi kabel serial kecepatan tinggi melintasi dua pasang kabel konduktor. Sebaliknya, paralel ATA menggunakan 16 kabel konduktor data yang masing-masing beroperasi pada kecepatan yang jauh lebih

rendah. SATA menawarkan beberapa kelebihan menarik dibandingkan antarmuka paralel ATA (PATA) yang lebih tua: mengurangi kabel yang tebal dan mengurangi biaya (berkurang dari delapan puluh kabel ke tujuh), lebih cepat dan lebih efisien dalam hal transfer data, dan fitur *hot swapping*^[1].

Pada 2009, SATA telah menggantikan sebagian luhur paralel ATA di semua konsumen pengiriman PC. PATA tetap digunakan dalam industri dan aplikasi benam (embedded application) yang tergantung pada penyimpanan CompactFlash, meskipun standar penyimpanan CFast yang baru akan didasarkan pada SATA



Gambar 2.13. **Buses Serial ATA, SATA**

Bus: USB

USB (Universal Serial Bus) adalah bus eksternal baru yang dikembangkan oleh Intel, Compaq, DEC, IBM, Microsoft, NEC dan Northern Telecom dan dirilis ke publik pada tahun 1996 dengan Mother Board Intel 430HX Triton II. USB memiliki kemampuan mentransfer 12 Mbps, mendukung hingga 127 perangkat dan hanya menggunakan satu IRQ. Agar komputer PC dapat memanfaatkan USB, pengguna harus menjalankan Windows 95 OSR2, Windows 98 atau Windows 2000. Pengguna Linux juga memiliki kemampuan menjalankan USB dengan driver dukungan yang tepat diinstal. Kabel USB adalah hot swappable yang memungkinkan pengguna untuk menghubungkan dan memutuskan kabel saat komputer menyala tanpa kerusakan fisik pada kabel.



Gambar 2.15. USB Logo

USB VERSIONS:

USB 1.0 - The original release of USB supports 127 devices transferring 12 Mbps.

USB 1.1 - Also known as full-speed USB, USB 1.1 is similar to the original release of USB however minor modifications for the hardware and the specifications. This version of USB still only supports a rate of 12 Mbps.

USB 2.0 - USB 2.0 also known as hi-speed USB was developed by Compaq, Hewlett Packard, Intel, Lucent, Microsoft, NEC and Philips and was introduced in 2001. Hi-speed USB is capable of supporting a transfer rate of up to 480 Mbps and is backwards compatible meaning it is capable of supporting USB 1.0 and 1.1 devices and cables.

USB 3.0

USB 3.0 was released in November 2008. The standard specifies a maximum transmission speed of up to 5 Gbit/s (625 MB/s), which is more than 10 times as fast as USB 2.0 (480 Mbit/s, or 60 MB/s), although this speed is typically only achieved using powerful professional grade or developmental equipment. USB 3.0 reduces the time required for data transmission, reduces power consumption, and is backward compatible with USB 2.0.

VERSI USB:

USB 1.0 - Rilis asli USB mendukung 127 perangkat yang mentransfer 12 Mbps.

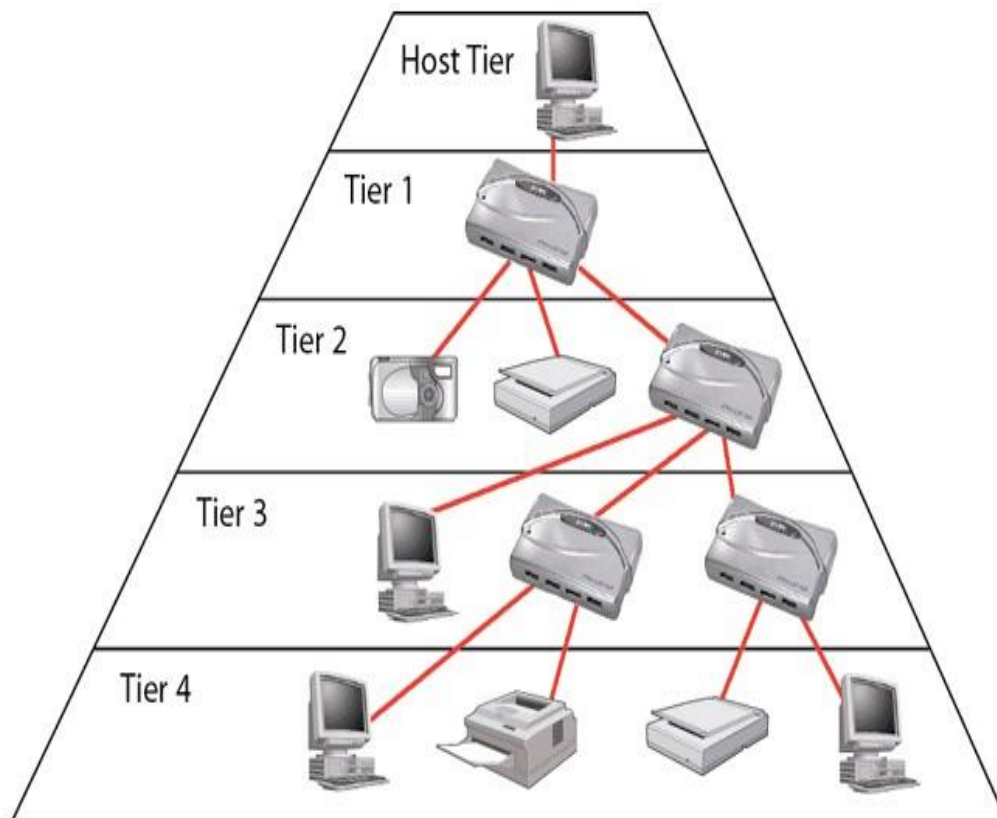
USB 1.1 - Juga dikenal sebagai USB kecepatan penuh, USB 1.1 mirip dengan rilis asli USB namun sedikit modifikasi untuk perangkat keras dan spesifikasinya. Versi USB ini masih hanya mendukung kecepatan 12 Mbps.

USB 2.0 –

USB 2.0 juga dikenal sebagai USB berkecepatan tinggi dikembangkan oleh Compaq, Hewlett Packard, Intel, Lucent, Microsoft, NEC dan Philips dan diperkenalkan pada tahun 2001. USB berkecepatan tinggi mampu mendukung kecepatan transfer hingga 480 Mbps dan kompatibel ke belakang yang berarti mampu mendukung perangkat dan kabel USB 1.0 dan 1.1.

USB 3.0

USB 3.0 dirilis pada November 2008. Standar menetapkan kecepatan transmisi maksimum hingga 5 Gbit/s (625 MB/s), yang lebih dari 10 kali lebih cepat dari USB 2.0 (480 Mbit/s, atau 60 MB/s), meskipun kecepatan ini biasanya hanya dicapai dengan menggunakan peralatan pengembangan atau tingkat profesional yang kuat. USB 3.0 mengurangi waktu yang diperlukan untuk transmisi data, mengurangi konsumsi daya, dan kompatibel dengan USB 2.0.



Gambar 2.16. Topologi HUB USB

Perangkat USB:

HUB

Menyederhanakan Konektivitas USB

Deteksi pasang dan lepas

Fungsi Perangkat USB yang mengirim atau menerima data

Bus: FireWire

Oleh Apple

Kecepatan

400Mbps

800 Mbps untuk 1394b

Dapat mengirim lebih dari satu CD setiap 10 detik

Pasang & mainkan Mendukung 63 perangkat

Memberikan kekuatan

Audio digital, video, hard drive eksternal, ... FireWire

Port FireWire asli lebih cepat dari USB saat diluncurkan. Kecepatan transfer hingga 400 Mbps. Jarak maksimum antar perangkat adalah 4,5 meter dari panjang kabel. Akhirnya, FireWire 800 menggantikan USB 2.0 dengan sangat mudah. FireWire 800 memiliki kecepatan transfer hingga 800 Mbps. Jarak maksimum panjang kabel antar perangkat adalah 100 meter.



Gambar 2.17. FireWire poart

Perbandinga antara FireWire dengan USB

Tabel 2.1. Perbandingan USB dengan FireWire

	USB	FireWire
On-bus power	2.5W	45W (!)
Max # devices	127	63
Topology	Star	Tree
Plug & Play	Yes	Yes
Peer-to-peer connectivity	No	Yes
Device Cost	Low	High

BUS: Bluetooth

Bluetooth adalah standar teknologi nirkabel terbuka eksklusif untuk pertukaran data jarak pendek (menggunakan transmisi radio gelombang pendek di pita ISM dari 2400-2480 MHz) dari perangkat tetap dan seluler, menciptakan jaringan area pribadi (PAN) dengan tingkat keamanan yang tinggi. Dibuat oleh vendor telekomunikasi Ericsson pada tahun 1994, pada awalnya dirancang sebagai alternatif nirkabel untuk kabel data RS-232. Itu dapat menghubungkan beberapa perangkat, mengatasi masalah sinkronisasi. Bluetooth dikelola oleh Bluetooth Special Interest Group, yang memiliki lebih dari 15.000 perusahaan anggota di bidang telekomunikasi, komputasi, jaringan, dan elektronik konsumen. SIG mengawasi pengembangan spesifikasi, mengelola program kualifikasi, dan melindungi merek dagang. Untuk dipasarkan sebagai perangkat Bluetooth, perangkat tersebut harus memenuhi standar yang ditentukan oleh SIG. Jaringan paten diperlukan untuk menerapkan teknologi dan hanya dilisensikan untuk perangkat yang memenuhi syarat tersebut; sehingga protokol, sementara terbuka, dapat dianggap sebagai milik Identifikasi frekuensi radio (RFID) adalah penggunaan sistem non-kontak nirkabel yang

menggunakan medan elektromagnetik frekuensi radio untuk mentransfer data dari tag yang terpasang pada suatu objek, untuk tujuan identifikasi dan pelacakan otomatis.

Beberapa tag tidak memerlukan baterai dan ditenagai oleh medan elektromagnetik yang digunakan untuk membacanya. Lainnya menggunakan sumber daya lokal dan memancarkan gelombang radio (radiasi elektromagnetik pada frekuensi radio). Tag berisi informasi yang disimpan secara elektronik yang dapat dibaca hingga beberapa meter (meter). Tidak seperti kode batang, tag tidak perlu berada dalam garis pandang pembaca dan dapat disematkan di objek yang dilacak. Tag RFID digunakan di banyak industri. RFID yang dipasang pada mobil selama produksi dapat digunakan untuk melacak kemajuannya melalui jalur perakitan. Farmasi dapat dilacak melalui gudang. Ternak dan hewan peliharaan mungkin memiliki tag yang disuntikkan, memungkinkan identifikasi positif dari hewan tersebut. Kartu identitas RFID dapat memberikan akses karyawan ke area bangunan yang terkunci, dan transponder RF yang dipasang di mobil dapat digunakan untuk menagih pengendara untuk akses ke jalan tol atau tempat parkir. Karena tag RFID dapat dilampirkan pada pakaian, harta benda, atau bahkan ditanamkan di dalam diri orang, kemungkinan membaca informasi yang terhubung secara pribadi tanpa persetujuan telah menimbulkan masalah privasi.

BUS: NFC berdasarkan RFID

Near field communication (NFC) adalah seperangkat standar untuk ponsel pintar dan perangkat serupa untuk membangun komunikasi radio satu sama lain dengan menyentuhnya bersama-sama atau membawanya ke jarak dekat, biasanya tidak lebih dari beberapa sentimeter. Aplikasi yang ada dan yang diantisipasi termasuk transaksi nirsentuh, pertukaran data, dan pengaturan komunikasi yang lebih kompleks yang disederhanakan seperti Wi-Fi. Komunikasi juga dimungkinkan antara perangkat NFC dan chip NFC tanpa daya, yang disebut "tag". Standar NFC mencakup protokol komunikasi dan format pertukaran data, dan didasarkan pada standar identifikasi frekuensi radio (RFID) yang ada termasuk ISO/IEC 14443 dan FeliCa. Standar tersebut termasuk ISO/IEC 18092 dan yang ditentukan oleh Forum NFC, yang didirikan pada tahun 2004 oleh Nokia, Philips dan Sony, dan sekarang memiliki 150 anggota. Forum juga mempromosikan NFC dan mengesahkan kepatuhan perangkat. NFC dibangun di atas sistem RFID dengan memungkinkan komunikasi dua arah antara titik akhir, di mana sistem sebelumnya seperti kartu pintar tanpa kontak hanya satu arah. Karena "tag" NFC

yang tidak berdaya juga dapat dibaca oleh perangkat NFC, ia juga mampu menggantikan aplikasi satu arah sebelumnya

Mikrokontroler

Mikrokontroler (MCU untuk unit mikrokontroler) adalah komputer kecil pada chip sirkuit terpadu (IC) VLSI tunggal. Mikrokontroler berisi satu atau lebih CPU (inti prosesor) bersama dengan memori dan periferan input/output yang dapat diprogram. Memori program berupa RAM feroelektrik, flash NOR atau ROM OTP juga sering disertakan dalam chip, serta sejumlah kecil RAM. Mikrokontroler dirancang untuk aplikasi tertanam, berbeda dengan mikroprosesor yang digunakan pada komputer pribadi atau aplikasi tujuan umum lainnya yang terdiri dari berbagai chip diskrit.

Dalam terminologi modern, mikrokontroler mirip dengan, tetapi kurang canggih dari, sistem pada chip (SoC). SoC dapat menghubungkan chip mikrokontroler eksternal sebagai komponen motherboard, tetapi SoC biasanya mengintegrasikan periferan canggih seperti unit pemrosesan grafis (GPU) dan pengontrol antarmuka Wi-Fi sebagai rangkaian unit mikrokontroler internalnya.

Mikrokontroler digunakan dalam produk dan perangkat yang dikontrol secara otomatis, seperti sistem kontrol mesin mobil, perangkat medis implan, kendali jarak jauh, mesin kantor, peralatan, perkakas listrik, mainan, dan sistem tertanam lainnya. Dengan mengurangi ukuran dan biaya dibandingkan dengan desain yang menggunakan mikroprosesor, memori, dan perangkat input/output terpisah, mikrokontroler membuatnya ekonomis untuk mengontrol lebih banyak perangkat dan proses secara digital. Mikrokontroler sinyal campuran adalah umum, mengintegrasikan komponen analog yang diperlukan untuk mengontrol sistem elektronik non-digital. Dalam konteks internet of things, mikrokontroler adalah sarana pengumpulan data yang ekonomis dan populer, merasakan dan menggerakkan dunia fisik sebagai perangkat tepi.

Seperti disebutkan sebelumnya, mikrokontroler seperti komputer kecil pada sebuah chip. Dan, seperti halnya komputer ukuran penuh, berbagai elemen membentuk bagian penting dari mikrokontroler dan membantu fungsinya. Mari kita bahas secara singkat masing-masing elemen ini.

UNIT PENGOLAHAN TENGAH (CPU)

CPU adalah inti dari mikrokontroler, sering disebut sebagai otak. Dibutuhkan input, menafsirkannya, dan menjalankan operasi, semua sesuai dengan program di memori. Ia bekerja bersama-sama dengan RAM dan ROM untuk menghasilkan output yang diinginkan dengan cepat dan dengan efisiensi tinggi.

MEMORI AKSES RANDOM (RAM)

Kita bisa menyebut RAM sebagai memori data. Ini adalah memori di mana data sementara disimpan saat CPU menjalankan operasi. Anda dapat menganggapnya seperti meja kerja tempat segala sesuatu diletakkan saat Anda mengerjakan elemen individu satu per satu.

MEMORI HANYA BACA (ROM)

ROM adalah memori program dalam mikrokontroler, menyimpan semua instruksi program. CPU mengambil instruksi dari ROM dan mengeksekusinya satu per satu. Tidak seperti RAM, ROM selalu menyimpan data bahkan setelah daya perangkat dimatikan. Menggunakan analogi sebelumnya, ini adalah lemari bengkel Anda, dimana barang-barang disimpan secara permanen, termasuk cetak biru Anda!

PORT INPUT/OUTPUT (I/O)

Kita dapat menggambarkan port I/O sebagai input sensorik dan anggota badan mikrokontroler. Ini menggunakan port ini untuk komunikasi dan antarmuka dengan perangkat eksternal. Penambahan seperti sakelar, termistor, akselerometer, dan sebagainya adalah input, sedangkan speaker, lampu, atau sinyal yang dikirim ke motor adalah contoh output.

PENGATUR WAKTU (Timer) & PENGHITUNG (counter)

Timer dan counter adalah elemen yang menemukan aplikasi yang paling bervariasi di mikrokontroler. Mereka dapat melakukan tugas-tugas seperti mengendalikan LED dan memodulasi kecepatan motor, dan CPU menjalankan operasi bersama-sama dengan jam mikrokontroler (sinyal yang terus-menerus, berkala, dan bergantian). Ini memungkinkan

fungsi seperti modulasi lebar pulsa (PWM), kontrol jam, pengukuran frekuensi, dan penghitungan pulsa eksternal.

KONVERTER ANALOG KE DIGITAL (ADC)

Seperti namanya, ADC mengubah sinyal input analog dari sensor eksternal menjadi sinyal digital. Hal ini diperlukan karena CPU membutuhkan input digital untuk melakukan operasi. Dengan cara ini, ADC membentuk jembatan antara input analog (misalnya tegangan yang diturunkan dari suhu) dan CPU.

KONVERTER DIGITAL KE ANALOG (DAC)

Ini adalah kebalikan dari ADC; DAC mengubah sinyal digital menjadi output analog murni. Ini kemudian lebih lanjut digunakan untuk mengontrol perangkat eksternal. Sebagai contoh, motor DC membutuhkan input analog, sehingga output digital dari CPU terlebih dahulu diubah menjadi output analog dan kemudian diteruskan ke motor.

Ada berbagai jenis mikrokontroler yang diklasifikasikan dengan cara yang berbeda. Kita dapat mengklasifikasikannya berdasarkan jumlah bit, perangkat memori, set instruksi, dan arsitektur. Namun, semua kategori ini berasal dari tiga jenis utama mikrokontroler.

8051

Mikrokontroler 8051 dari Intel adalah salah satu mikrokontroler yang paling umum digunakan, bahkan hingga saat ini. Diciptakan pada 1980-an, mikrokontroler ini membuka jalan bagi masa depan komputasi mikro. Ini menampilkan CPU 8-bit, dengan 64 KB ROM dan ruang alamat RAM 128 byte.

Karena didasarkan pada arsitektur Harvard, memori program dan memori data tetap terpisah. Ini memberikan kecepatan komputasi yang cepat dan menghindari tumpang tindih. Mikrokontroler 8051 ditemukan di banyak aplikasi karena kemampuannya untuk diintegrasikan ke dalam proyek yang lebih kecil dengan mudah.

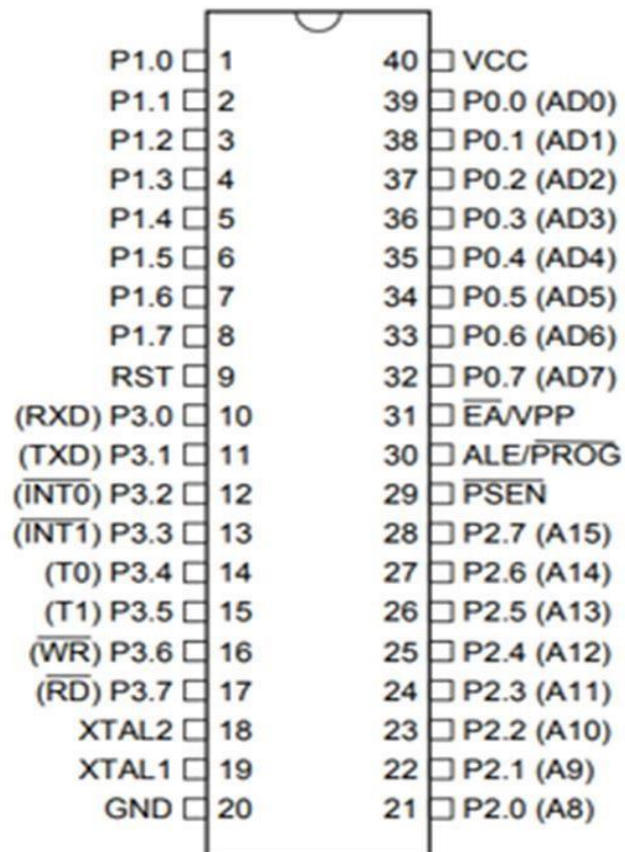
AVR

Mikrokontroler AVR adalah chip yang akan Anda temui setidaknya sekali jika Anda menyukai elektronik. Mikrokontroler ini digunakan di banyak papan pengembangan,

termasuk Arduino yang terkenal. Arsitekturnya sama dengan 8051 tetapi dengan peningkatan yang mencolok. AVR didasarkan pada set instruksi RISC, yang memungkinkan mereka untuk menghitung pada kecepatan yang jauh lebih tinggi sambil tetap mengkonsumsi lebih sedikit daya daripada 8051.

PIC adalah singkatan dari peripheral interface controller, sejenis mikrokontroler yang pertama kali dikembangkan pada tahun 1975 dan terutama digunakan untuk aplikasi industri. Ini juga didasarkan pada arsitektur Harvard dan set instruksi tipe RISC, sedikit meningkatkan mikrokontroler tipe AVR. PIC juga memiliki set instruksi yang lebih kecil daripada AVR, sehingga membuatnya lebih mudah dipelajari. Di antara beberapa mikrokontroler PIC yang sangat terkenal adalah PIC18fXX8, PIC16f88X, dan PIC32MXX.

Mikrokontroler AT 89C51

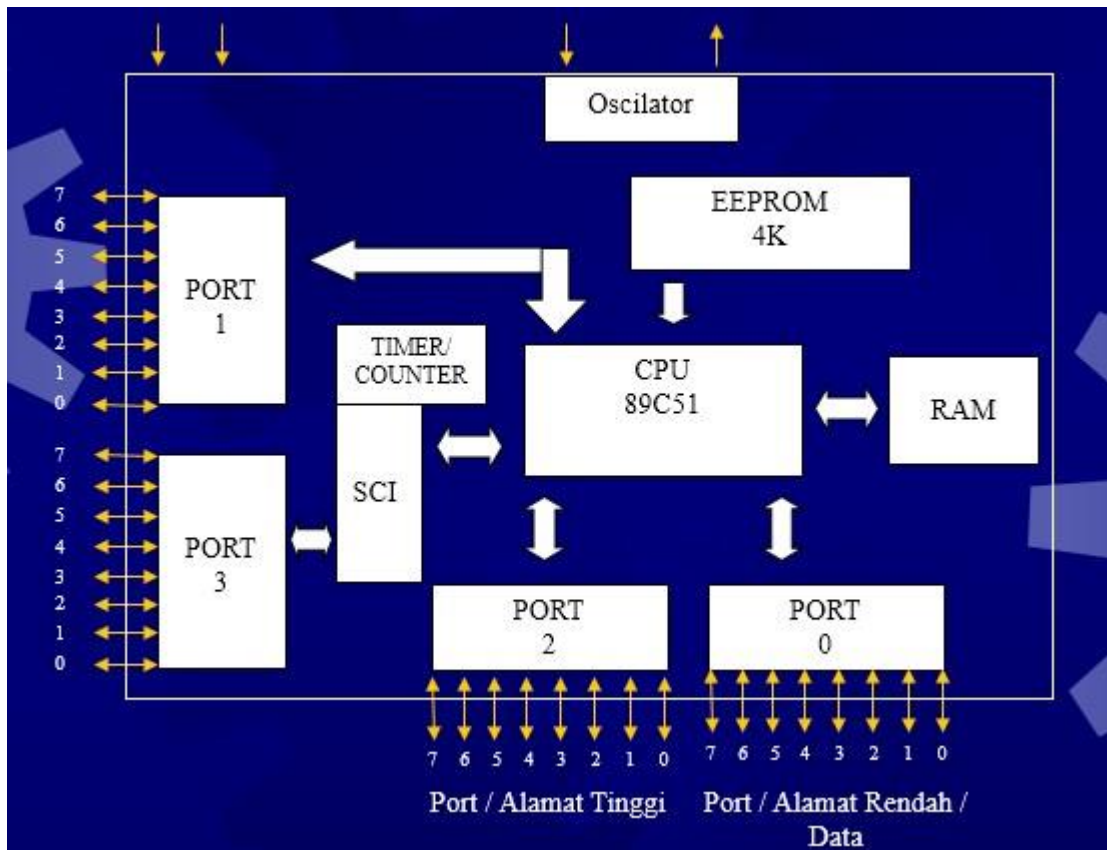


Gambar 2.18 Pin Mikrokontroler AT 89C51

1	P1.0	Pin ke-0 PORT P1
2	P1.1	Pin pertama PORT P1
3	P1.2	Pin ke-2 PORT P1
4	P1.3	Pin ke-3 PORT P1
5	P1.4	Pin ke-4 PORT P1
6	P1.5	Pin ke-5 PORT P1
7	P1.6	Pin ke-6 PORT P1
8	P1.7	Pin ke-7 PORT P1
9	RST	Atur ulang pin Mikrokontroler
10	(RXD)	P3.0 Pin ke-0 PORT P3 atau pin Penerima Mikrokontroler
11	(TXD)	P3.1 Pin 1 PORT P3 atau pin Pemancar Mikrokontroler
12	(INT0)	P3.2 Pin ke-2 PORT P3 atau Interupsi Eksternal 0 dari MCU
13	(INT1)	P3.3 Pin ke-3 PORT P3 atau Interupsi Eksternal 1 dari MCU
14	(T0)	P3.4 Pin ke-4 PORT P3 atau Timer 0 interupsi MCU
15	(T1)	P3.5 Pin ke-5 PORT P3 atau interupsi Timer 1 dari MCU
16	(WR)	P3.6 Pin ke-6 PORT P3 atau Tulis ke Pin memori data eksternal
17	(RD)	P3.7 Pin ke-7 PORT P3 atau Baca dari Pin memori data eksternal
18	XTAL2	Pin kristal eksternal 2 dari Mikrokontroler
19	XTAL1	Pin kristal eksternal 1 dari Mikrokontroler
20	GND	Pin ground MCU
21	P2.0 (A8)	Pin ke-0 dari PORT P2 atau Alamat Orde Tinggi bit 8 dari MCU
22	P2.1(A9)	Pin 1 PORT P2 atau Alamat Orde Tinggi bit 9 dari MCU
23	P2.2(A10)	Pin ke-2 dari PORT P2 atau Alamat Orde Tinggi bit 10 dari MCU
24	P2.3(A11)	Pin ke-3 dari PORT P2 atau Alamat Orde Tinggi bit 11 dari MCU
25	P2.4(A12)	Pin ke-4 PORT P2 atau Alamat orde tinggi bit 12 dari MCU
26	P2.5(A13)	Pin ke-5 PORT P2 atau Alamat orde tinggi bit 13 dari MCU
27	P2.6(A14)	Pin ke-6 PORT P2 atau Alamat orde tinggi bit 14 dari MCU
28	P2.7(A15)	Pin ke-7 dari PORT P2 atau Alamat Orde Tinggi bit 15 dari MCU
29	PSEN	Pin aktifkan penyimpanan program, Baca memori program eksternal
30	ALE/PROG EA/VPP	Address Latch Enable/ Program input Pulse untuk flashing 31 Akses Aktifkan tegangan/Program mengaktifkan tegangan
32	P0.7(AD7)	Pin ke-7 dari PORT P0 atau Alamat orde rendah bit 7 dari MCU
33	P0.6(AD6)	Pin ke-6 PORT P0 atau Alamat orde rendah bit 6 MCU
34	P0.5(AD5)	Pin ke-6 PORT P0 atau Alamat orde rendah bit 5 dari MCU
35	P0.4(AD4)	Pin ke-6 PORT P0 atau Alamat orde rendah bit 4 MCU
36	P0.3(AD3)	Pin ke-3 PORT P0 atau Alamat orde rendah bit 3 MCU
37	P0.2(AD2)	Pin ke-2 PORT P0 atau Alamat orde rendah bit 2 dari MCU
38	P0.1(AD1)	Pin pertama PORT P0 atau Alamat orde rendah bit 1 MCU
39	P0.0(AD0)	Pin ke-0 PORT P0 atau Alamat orde rendah bit 0 dari MCU
40	Vcc	

Pasokan pin MCU Berapa Frekuensi Kerja ?

- Mikrokontroler 89C51 sudah dilengkapi dengan osilator di dalam chip, dengan frekuensi kerja antara 1,2 MHz – 12 MHz.
- Dapat juga digunakan kristal atau pembangkit Clock luar (external) Bagaimana Arsitektur 89C51 ?



Gambar 2.19. Arsitektur 89C51

Apakah Chip 89C51 ini punya EPROM Internal ?

- Prosesor 89C51 mempunyai EPROM internal, sehingga penyimpanan program dapat dilakukan pada EPROM ini.
- Chip ini punya RAM internal yang terbagi ke dalam 3 kategori register, yaitu :
 - Register Khusus
 - Register Serba Guna

- ✱ Register Kelompok
 - Port P0 dan P2 digunakan untuk digunakan untuk pengaksesan memori eksternal.
 - P0 digunakan untuk mengeluarkan alamat bagian bawah yang di multipleks dengan bus data
 - P2 digunakan untuk mengeluarkan alamat memori bagian atas. Sehingga lebar alamat memori adalah 16 bit

P1 berfungsi sebagai port biasa

- P3 selain berfungsi sebagai port biasa, dapat juga berfungsi sebagai :
 - P3.0 : port input serial (RxD)
 - P3.1 : port output serial (TxD)
 - P3.2 : input interupsi eksternal 0 (INT 0)
 - P3.3 : input interupsi eksternal 1 (INT 1)
 - P3.4 : input eksternal untuk pewaktu/pencacah 0 (T0)
 - P3.5 : input eksternal untuk pewaktu/pencacah 1 (T1)
 - P3.6 : sinyal tulis memori eksternal (WR)
 - P3.7 : sinyal baca memori eksternal (RD) Cara Memprogram Mikrokontroler AT89C51 (8051)

Mikrokontroler Atmel dapat diprogram dengan berbagai software yang tersedia di pasaran. Arduino, Keil uVision adalah platform yang paling banyak digunakan untuk beberapa nama. Jika Anda berencana membuat program serius dan ekspansi dengan dukungan komunitas, maka Keil direkomendasikan.

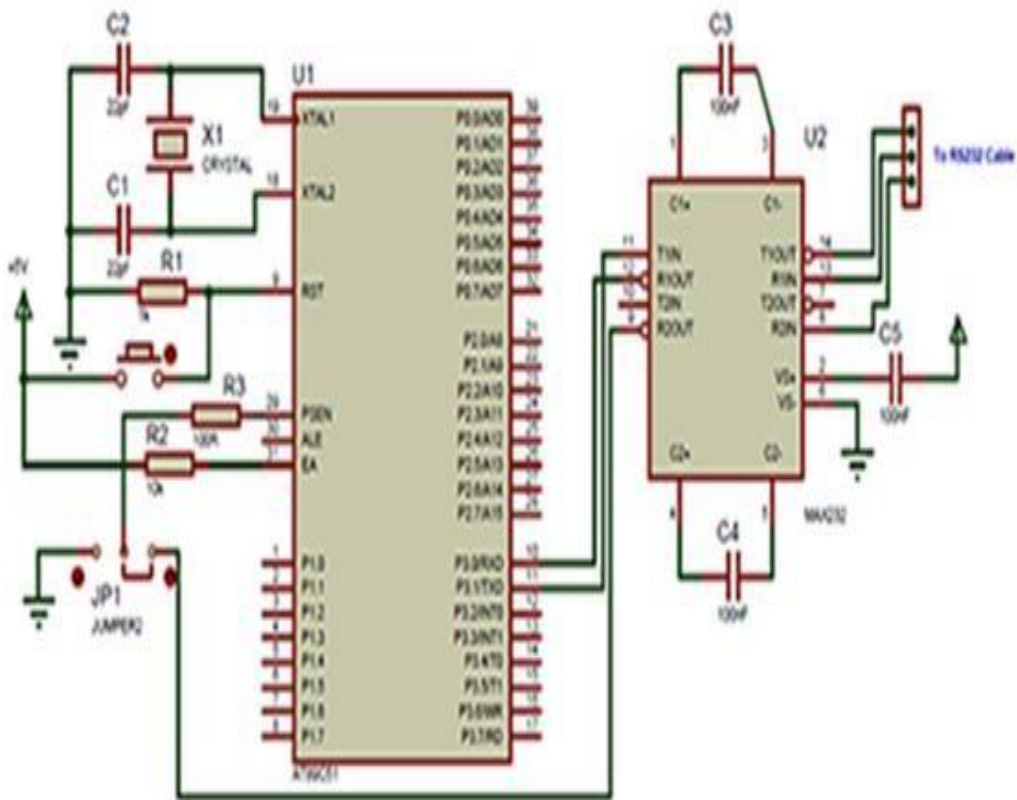
Untuk memprogram mikrokontroler Atmel kita memerlukan IDE (Integrated Development Environment), tempat pemrograman berlangsung. Kompilator, tempat program kami diubah menjadi bentuk yang dapat dibaca MCU yang disebut file HEX. IPE (Integrated Programming Environment), yang digunakan untuk membuang file hex kami ke MCU kami.

IDE	: Keil uVision IDE
Perangkat Keras Pemrograman	: MAX232 dengan konverter RS232 ke USB
Pemrogram	: USBASP

Untuk membuang atau mengunggah kode kita ke dalam IC Atmel kita membutuhkan seorang programmer, programmer yang paling umum digunakan adalah USBASP yang

harus dibeli secara terpisah. Juga mensimulasikan program Anda pada perangkat lunak sebelum mencobanya pada perangkat keras akan menghemat banyak waktu. Jadi Anda dapat menggunakan perangkat lunak seperti ISIS proteus dari Labcenter untuk mensimulasikan program Anda.

Rangkaian Pemrograman 8051 (AT89C51) ditunjukkan di bawah ini:



Gambar 2.19. Wiring diagram ISP dengan mikrokontroler

BAB III

Port I/O dan system clock Mikrokomputer

Mata Kuliah : TEKNIK INTERFACE

Capaian Pembelajaran :

Mahasiswa Mampu menerapkan dan mengembangkan teknik antarmuka, dalam sistem mikrokomputer sesuai dengan kebutuhan desain yang harus dipenuhi Kemampuan

Akhir yang Diharapkan :

Memahami jenis-jenis Port I/O dan system clock Mikrokomputer

Alokasi Waktu : 3 X 50 menit

Pertemuan ke : 3 Indikator

:

1.

Mahasiswa akan mampu menjelaskan antarmuka yang disediakan oleh mikrokontroler AT89S52 (MCS51)

2. Mahasiswa akan mampu menjelaskan antarmuka yang disediakan oleh mikrokontroler AVR ATmega32P
3. Mahasiswa akan mampu mengaplikasikan rangkaian sumber daya yang stabil ke system mikrokomputer
4. Mahasiswa akan mampu mengaplikasikan rangkaian clock ke sistem mikrokomputer sesuai dengan kebutuhan dan konstrain kecepatan mikrokomputer
5. Mahasiswa akan mampu mengaplikasikan rangkaian reset ke system mikrokomputer dengan tepat

Dasar-dasar I/O

Apa Artinya Input/Output (I/O)?

Input/output (I/O), dalam komputasi, adalah proses komunikasi antara komputer dan dunia luar.

Pada tingkat yang paling dasar, sistem informasi (IS), seperti aplikasi perangkat lunak, diinstal pada komputer dan penggunanya di dunia luar mengoperasikan komputer untuk mendapatkan solusi untuk masalah. Input mengacu pada sinyal atau instruksi yang dikirim ke komputer. Output mengacu pada sinyal yang dikirim dari komputer.

Ada operasi input dan output yang dilakukan dalam konteks komputer di mana-mana.

Perangkat I/O umum sederhana termasuk mouse, keyboard, monitor, dan printer.

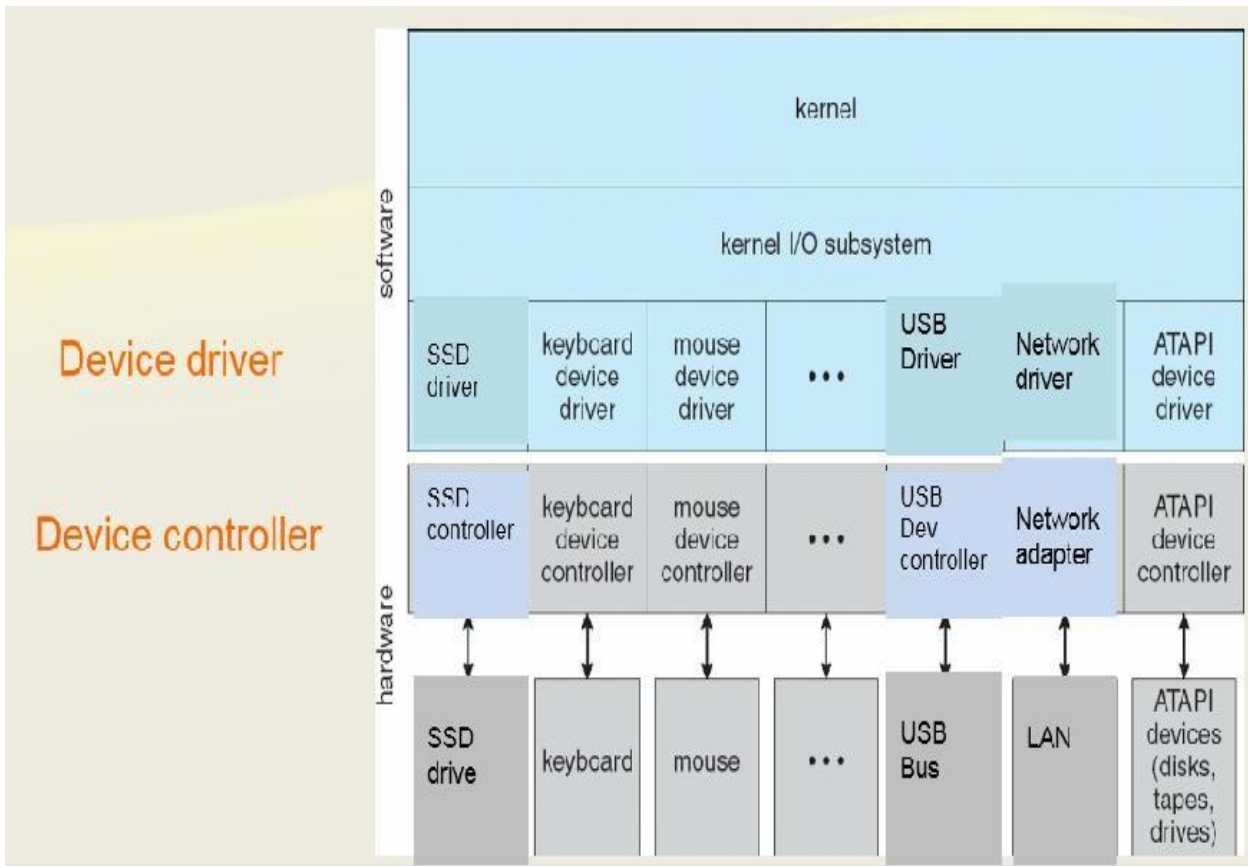
Berikut ini adalah konteks utama untuk istilah I/O:

Antarmuka I/O: Ini menyediakan cara untuk berinteraksi dengan perangkat keras komputer. Antarmuka setiap perangkat mampu menyandikan dan mendekode sinyal I/O dalam bentuk yang dapat dipahami baik untuk perangkat input maupun output.

I/O Aplikasi yang Dapat Diprogram: Banyak aplikasi terintegrasi dengan sistem operasi (OS) yang menyediakan input dan output waktu proses secara bersamaan. Contoh terbaik adalah aplikasi pemrograman C, C++ dan Java, yang memiliki pustaka bawaan yang digunakan untuk operasi I/O. Program ditulis sehingga satu file library digunakan sebagai input sedangkan output ditampilkan kepada pengguna. Dalam pemrograman, konsep ini dikenal sebagai penanganan file.

Memory Addressing I/O: Memori komputer berisi blok untuk menyimpan aplikasi/proses untuk diproses. Banyak mekanisme pengalamatan yang digunakan untuk tujuan ini; masing-masing menggunakan operasi I/O dalam beberapa konteks. Pengalamatan memori menggunakan operasi I/O memori adalah pengalamatan terindeks dan pengalamatan langsung.

Device driver



Gambar 3.1. hubungan antara software driver dan device hardware

Pengontrol I/O Port perangkat I/O Mentransfer data ke/dari perangkat Menyinkronkan operasi dengan perangkat lunak Register perintah / kontrol Register status: status perangkat, kesalahan register data Tulis: data perangkat mis. Kirim Baca: CPU perangkat misalnya menerima beberapa register I/O dapat dipetakan ke alamat yang sama Pengontrol Perangkat diperlukan I/O elektro-mekanis mekanik: perangkat itu sendiri elektronik: pengontrol/adaptor perangkat

Antarmuka tingkat rendah antara pengontrol dan perangkat Contoh: Pengontrol disk aliran bit serial dari drive blok byte, sektor, trek, koreksi kesalahan.

Perangkat karakter: input, port serial, suara, I2C

Blokir perangkat: penyimpanan USB, SSD, CD

Perangkat jaringan: LAN atau Wi-fi

Device Drivers

- ◆ Device-specific – controls IO device, written by manufacturer eg SSD driver, mouse, command registers, commands vary from device to device
- ◆ Part of kernel Compiled with OS, Dynamically loaded into OS
- ◆ Accept abstract R/W requests from device-independent layer ;
- ◆ Initialize device, Manage power, and log
- ◆ Check & translate input parameters

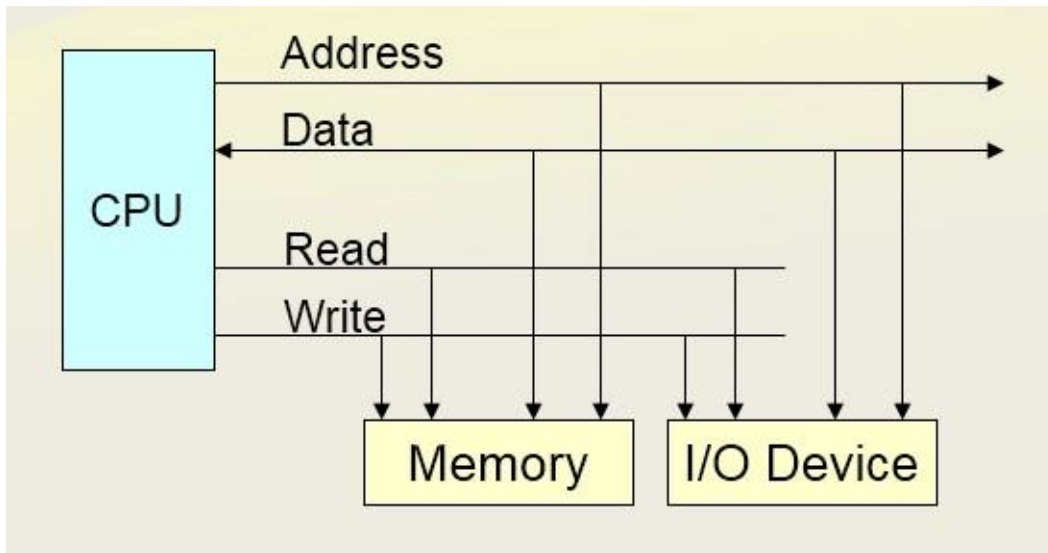
e.g., convert linear block number into the head, track, sector and cylinder number for disk access
 Check device status & Control → sequence of commands

Driver Perangkat perangkat Khusus – mengontrol perangkat IO, yang ditulis oleh pabrikan misalnya driver SSD, mouse, register perintah, perintah bervariasi dari perangkat ke perangkat Bagian dari kernel Dikompilasi dengan OS, Dimuat secara dinamis ke dalam OS

Terima permintaan R/W abstrak dari lapisan yang tidak bergantung pada perangkat;
 Inisialisasi perangkat, Kelola daya, dan log Periksa & terjemahkan parameter input misalnya, mengubah nomor blok linier menjadi nomor kepala, trek, sektor, dan silinder untuk akses disk Periksa status perangkat & Kontrol urutan perintah

CPU Bus I/O

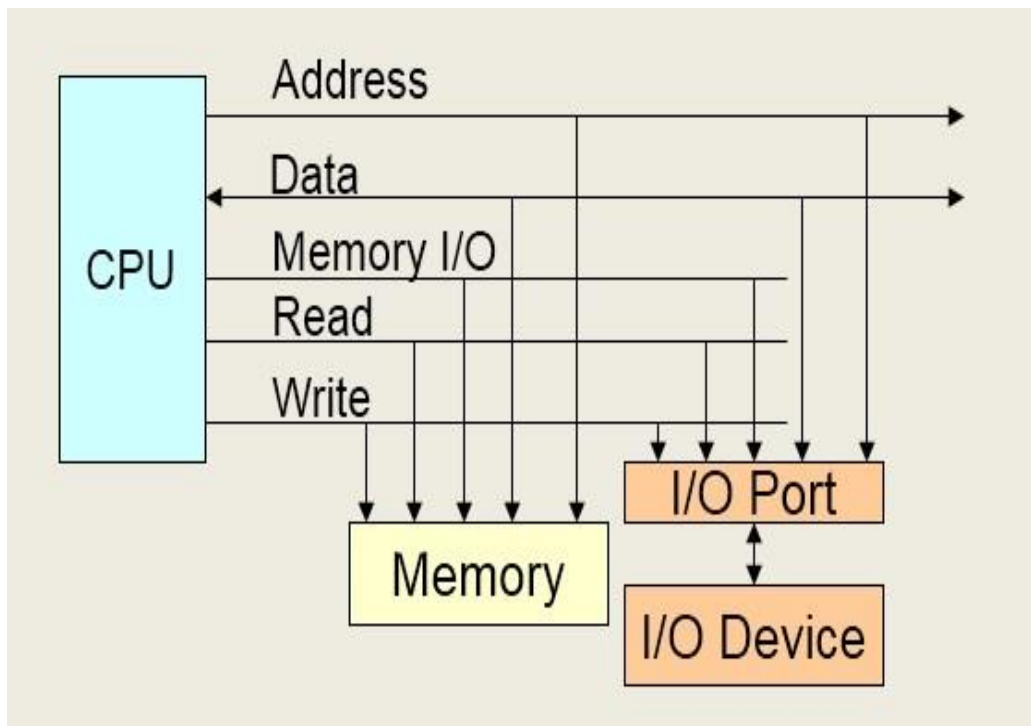
CPU perlu berkomunikasi dengan I/O Memori, I/O yang dipetakan Perangkat yang dipetakan ke lokasi memori yang dicadangkan - seperti RAM Menggunakan instruksi memuat/menyimpan seperti halnya akses ke memori



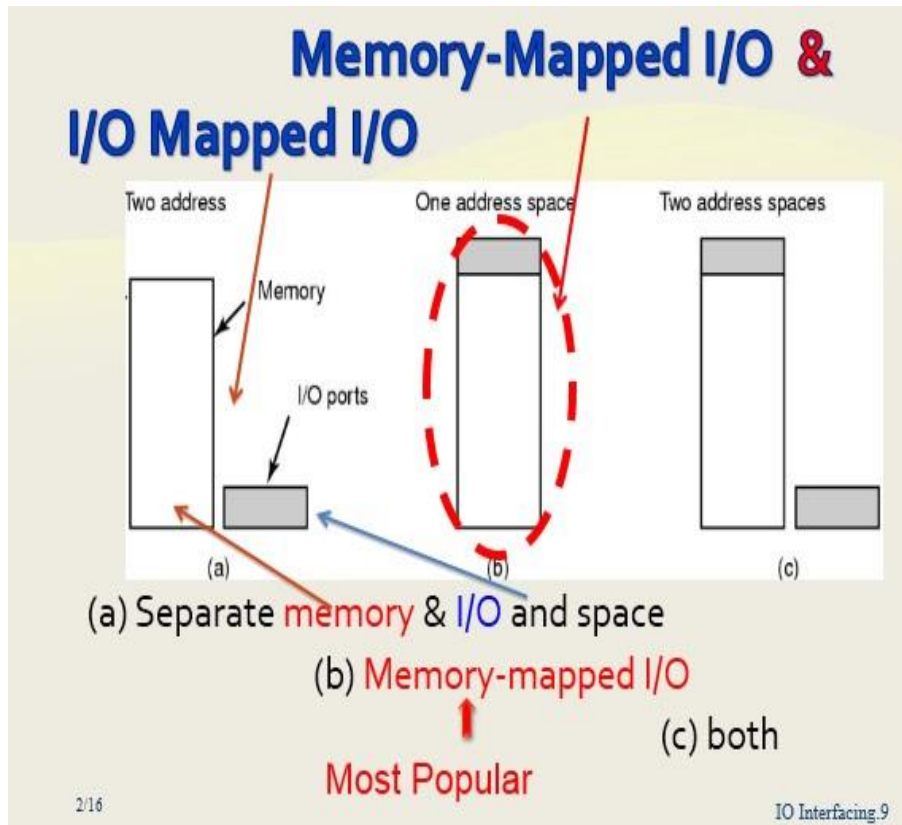
Gambar 3.2. Bus I/O

I/O mapped I/O

I/O yang dipetakan I/O Jalur bus khusus dan Instruksi khusus



Gambar 3.3. Jalur bus I/O khusus

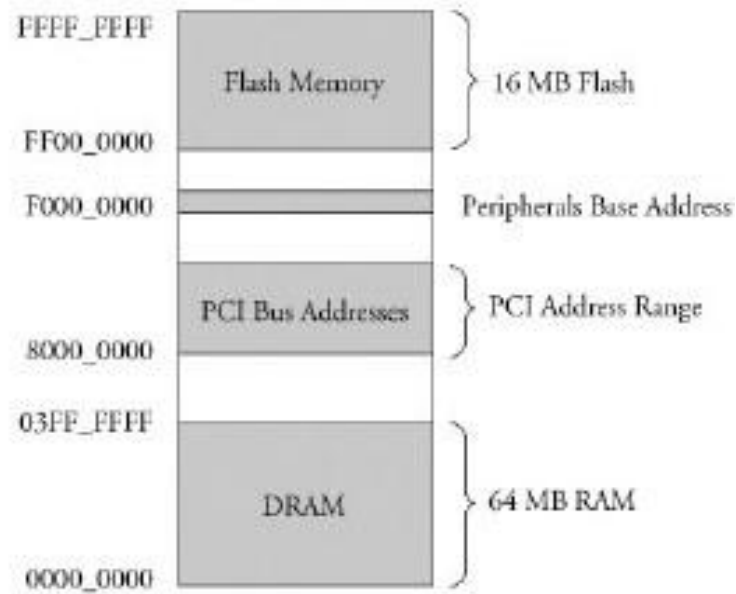


Gambar 3.4. Pemetaan I/O

Memori Sistem Tertanam

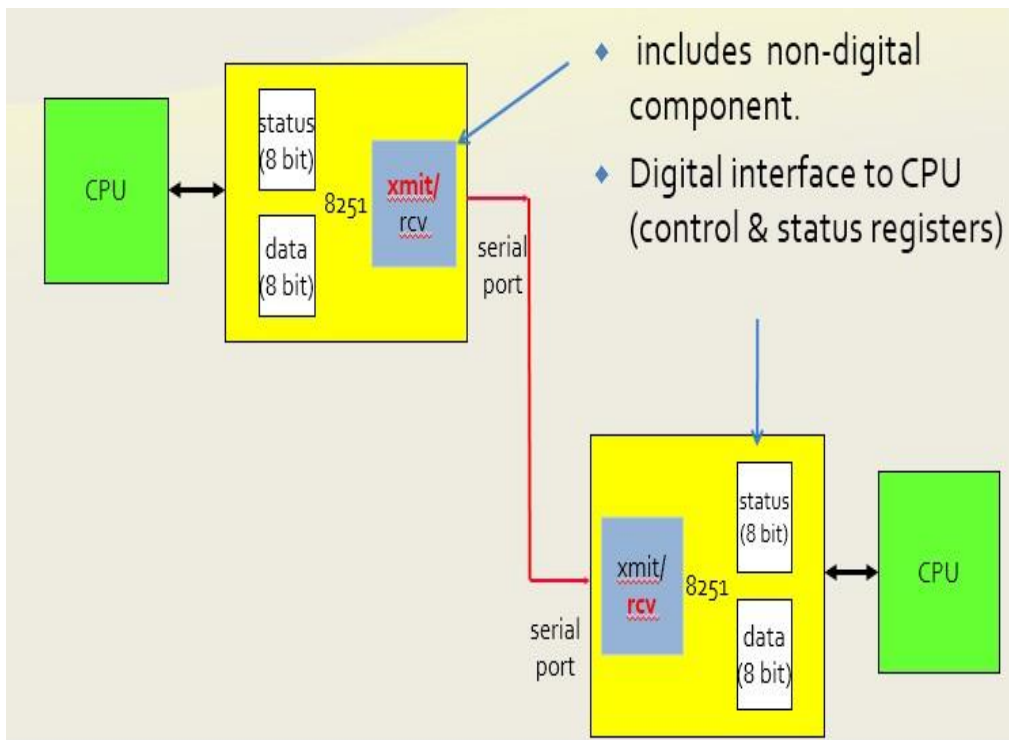
Memori Peta dan I/O Persyaratan Ruang Memori Aplikasi
memori flash, kernel, boot, ..

- DRAM
- Ruang cadangan I/O untuk periferal BIOS f \square mengontrol prosesor pada powerup, Inisialisasi perangkat keras, subsistem memori
- memuat OS dari drive
- alias bootloader



Gambar 3.5. Pemetaan Memory

I/O Minicom UART Serial I/O CPU interface



Gambar 3.6. Hubungan Interface serial

Teknik Transfer Data I/O

- polling I/O
- interrupt-driven I/O

CPU mentransfer data antara memori dan I/O Memakan waktu untuk perangkat berkecepatan tinggi

- Direct memory access (DMA)

Akses memori langsung (DMA) OS menyediakan info awal: alamat memori, kontrol Kontroler DMA mentransfer ke/dari memori secara mandiri - langsung

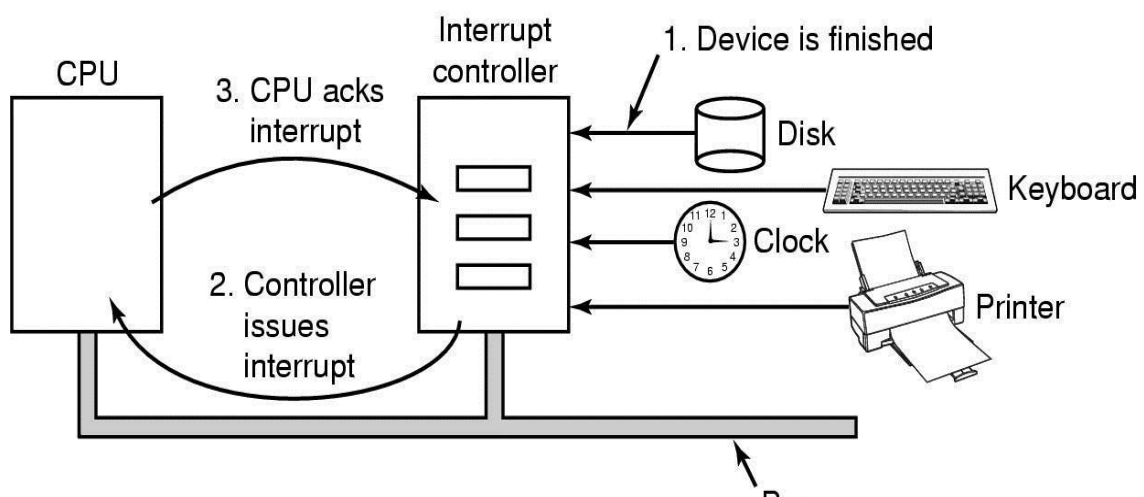
Pengontrol menyela saat penyelesaian / kesalahan

Mekanisme I/O – Polling Periksa status I/O secara berkala Jika perangkat siap, lakukan operasi Jika salah, ambil tindakan sistem tertanam real-time kecil atau kinerja rendah

Waktu yang dapat diprediksi Biaya perangkat keras rendah membuang waktu CPU

Mekanisme –Interrupts I/O

Metode I/O berbasis interupsi mengontrol aktivitas transfer data ke dan dari perangkat I/O yang terhubung. Hal ini memungkinkan CPU untuk terus memproses pekerjaan lain sebagai gantinya dan akan terganggu hanya ketika menerima sinyal input dari perangkat I/O. Misalnya, jika Anda menekan tombol pada keyboard, I/O interupsi akan mengirim sinyal ke CPU bahwa CPU perlu berhenti sejenak dari tugasnya saat ini dan menjalankan permintaan dari pukulan keyboard.



Gambar 3.7. Mekanisme –Interrupts I/O

Devices managed by interrupt controller → interrupt lines

Interrupts

- ◆ When device ready or error
Interrupt → CPU, checked every instruction
- ◆ CPU acknowledges interrupt ; saves the state interrupt
handler dispatched
determines cause, services the device and clears the interrupt
Context switch time
- ◆ Priority interrupts higher priority devices
get more attention Can interrupt
lower priority device

Interupsi

Saat perangkat siap atau error Interupsi CPU, periksa setiap instruksi CPU mengakui interupsi; menyelamatkan negara penanganan interupsi dikirim menentukan penyebab, melayani perangkat dan menghapus interupsi Waktu beralih konteks

Interupsi prioritas perangkat dengan prioritas lebih tinggi mendapatkan lebih banyak perhatian Dapat mengganggu perangkat dengan prioritas lebih rendah

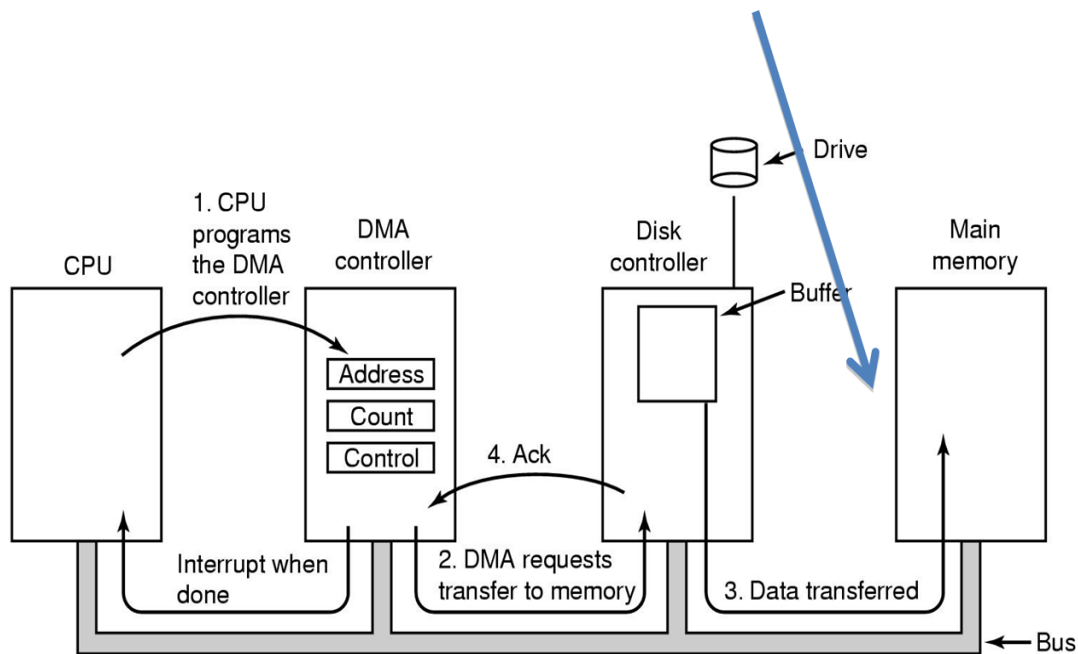
Mekanisme I/O DMA

Perangkat lambat seperti keyboard akan menghasilkan interupsi ke CPU utama setelah setiap byte ditransfer. Jika perangkat cepat seperti disk menghasilkan interupsi untuk setiap byte, sistem operasi akan menghabiskan sebagian besar waktunya untuk menangani interupsi ini. Jadi komputer biasa menggunakan perangkat keras akses memori langsung (DMA) untuk mengurangi overhead ini.

Direct Memory Access (DMA) berarti CPU memberikan otoritas modul I/O untuk membaca dari atau menulis ke memori tanpa keterlibatan. Modul DMA sendiri mengontrol pertukaran data antara memori utama dan perangkat I/O. CPU hanya terlibat pada awal dan akhir transfer dan terputus hanya setelah seluruh blok telah ditransfer.

Direct Memory Access membutuhkan perangkat keras khusus yang disebut DMA controller (DMAC) yang mengatur transfer data dan mengatur akses ke bus sistem. Kontroler diprogram dengan penunjuk sumber dan tujuan (tempat membaca/menulis

data), penghitung untuk melacak jumlah byte yang ditransfer, dan pengaturan, yang mencakup tipe I/O dan memori, interupsi, dan status untuk siklus CPU.



Gambar 3.8. Mekanisme I/O DMA

Operasi transfer DMA

Akses Memori Langsung (DMA)

Data Xfer: Perangkat I/O Mem OR I/O Mem

CPU membantu & memulai Xfer mendapatkan data dari pengontrol I/O ke CPU pada byte – basis tidak efisien – (misalnya transfer data disk) Alih-alih tujuan khusus Kontroler DMA Lebih cepat, lebih efisien

Protokol DMA-CPU mis. DISK baca

Program CPU DMA controller: - setup tentukan alamat sumber / tujuan jumlah byte dan informasi kontrol (mis., Baca/tulis)

Pengontrol DMA: Akui - Xfer

Memulai Xfer secara langsung tanpa bantuan CPU Minta pengontrol I/O untuk memindahkan data ke memor. Data pengontrol disk memori utama Pengontrol disk menyelesaikan transfer ke pengontrol DMA Jabat tangan antara pengontrol DMA dan pengontrol perangkat Pencurian siklus

Pengontrol DMA mencuri siklus bus dari CPU – memblokir CPU agar tidak mengakses memori DMA meningkatkan kinerja sistem

Memilih metode I/O

I/O terprogram

I/O yang digerakkan oleh interupsi I/O

menggunakan DMA

tercepat untuk satu permintaan I/O? throughput

tertinggi?

Contoh penggunaan dari masing-masing grup

I/O Buffer - menghaluskan ketidakcocokan kecepatan

Buffer:: area memori menyangga data antara perangkat dan aplikasi.

Alasan buffering: ketidakcocokan kecepatan antara produsen dan konsumen aliran data - gunakan buffering ganda. Perangkat memiliki ukuran transfer data yang berbeda aplikasi menulis ke buffer; Salinan OS ke buffer kernel dan disk.

Strategi input tanpa buffer tidak efektif

Kesalahan I/O harus Dilaporkan

Kesalahan IO Sebenarnya di tingkat perangkat (mis., blok disk rusak, kamera dimatikan)

Kesalahan Pemrograman IO misalnya, menulis ke keyboard, membaca dari printer).

perangkat lunak IO yang tidak bergantung pada perangkat mendeteksi / merespons kesalahan; laporan perangkat lunak IO ruang pengguna.

BAB IV

Sistem Minimum Mikrokontroler

Mata Kuliah : TEKNIK INTERFACE
Capaian Pembelajaran :

Mahasiswa Mampu menerapkan dan mengembangkan Sistem Minimum Mikrokontroler, dalam sistem mikrokomputer sesuai dengan kebutuhan desain yang harus dipenuhi Kemampuan Akhir yang Diharapkan :

Memahami komponen penyusun Sistem Minimum Mikrokontroler

Alokasi Waktu : 3 X 50 menit

Pertemuan ke : 2 Indikator

:

1. Mahasiswa akan mampu menjelaskan Sistem Minimum Mikrokontroler
2. Mahasiswa akan mampu menjelaskan komponen penyusun Sistem Minimum
3. Mahasiswa akan mampu mengaplikasikan rangkaian sumber daya yang stabil ke system mikrokomputer
4. Mahasiswa akan mampu mengaplikasikan rangkaian clock ke sistem mikrokomputer sesuai dengan kebutuhan dan konstrain kecepatan mikrokomputer
5. Mahasiswa akan mampu mengaplikasikan rangkaian reset ke system mikrokomputer dengan tepat

Sistem Minimum Mikrokontroler

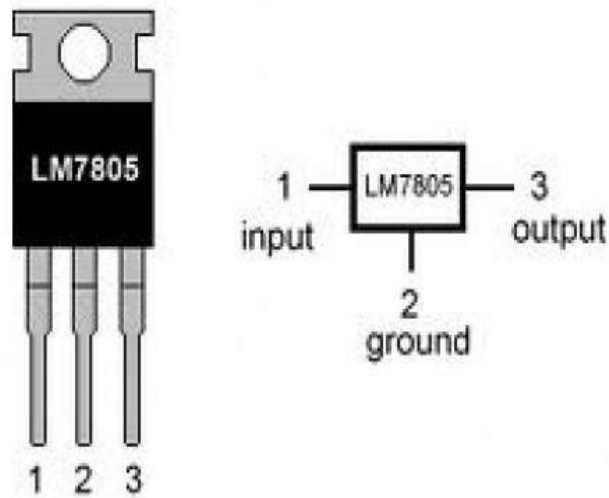
Sistem Minimum Mikrokontroler adalah sebuah rangkaian paling sederhana dari sebuah mikrokontroler agar IC mikrokontroler tersebut bisa beroperasi dan diprogram. Dalam aplikasinya sistem minimum sering dihubungkan dengan rangkaian lain untuk tujuan tertentu.

Sismin mikrokontroler ini memiliki pendukung input/output yang programmable dan RAM yang On-Chip. Sismin ini dapat dibuat sangat fleksibel tergantung aplikasi yang akan dibuat. Pada umumnya, suatu mikrokontroler membutuhkan dua elemen (selain power supply) untuk berfungsi: Kristal Oscillator (XTAL), dan Rangkaian RESET, 2 elemen tersebut merupakan syarat utama terbentuknya Sismin (Sistem Minimum). Ada beberapa yang harus diperhatikan dalam membuat sistem minimum mikrokontroler, yaitu:

Power Supply

Semua komponen elektronika membutuhkan power supply atau sering juga disebut catu daya. Mikrokontroler beroperasi pada tegangan 5 volt. Biasanya pembuatan catu daya mikrokontroler menggunakan IC regulator 7805 agar tegangannya bisa stabil.

LM7805 PINOUT DIAGRAM



Gambar 4.1. IC regulator 7805

Osilator (Pembangkit Frekuensi)

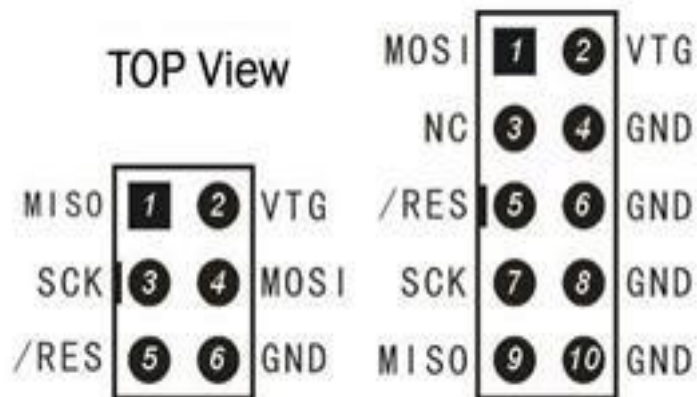
Pada dasarnya mikrokontroler memiliki sifat seperti manusia. Kalau manusia memiliki jantung untuk bisa hidup maka mikrokontroler memiliki osilator untuk bisa beroperasi. Mikrokontroler sendiri sudah memiliki osilator internal yaitu sebesar 8Mhz tetapi kadang kala agar kinerja mikronkontroler lebih cepat osilator internal tidak bisa menangani kasus tersebut. Oleh karena itu dibutuhkan osilator eksternal (kristal) yang nilainya lebih dari 8Mhz. Perlu diperhatikan mikrokontroler hanya bisa beroperasi sampai 16 Mhz. jadi kalau memilih krsital untuk AVR tidak boleh lebih dari 16Mhz.



Gambar 4.2. Kristal 16 MHz

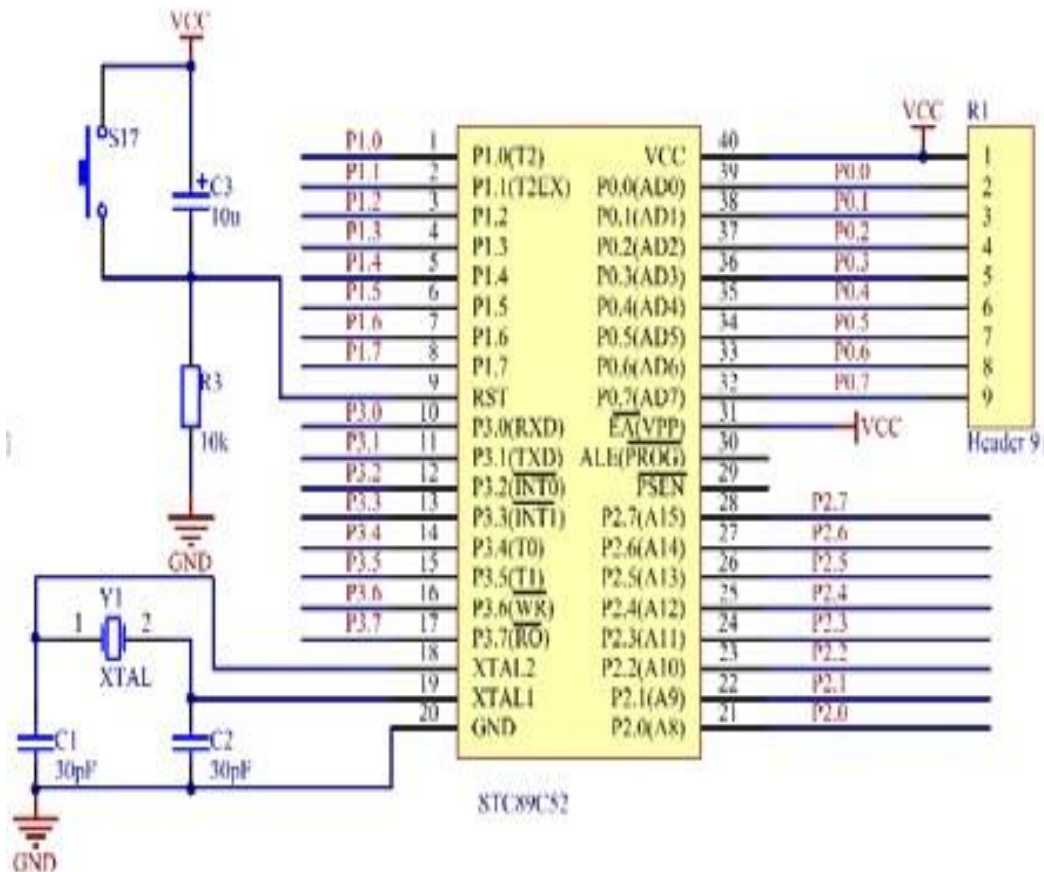
ISP (In-System Programmable)

Sistem Minimum Mikrokontroler dibuat untuk di program. Prinsipnya mikrokontroler bisa diprogram secara parallel atau secara seri. Pemrograman mikrokontroler secara seri atau lebih dikenal dengan ISP tidak perlu memerlukan banyak jalur data. Tapi ISP memiliki kelemahan, jika salah setting fuse bit yang memiliki fungsi fital misal pin reset di disable maka alamat DEH sudah tidak bisa digunakan lagi. Untuk mengembalikan settingan fuse bit tadi, harus menggunakan pemrograman tipe parallel (high voltage programming).



Gambar 4.3. : Settingan Port ISP

Berikut Gambar Rangkaian Sistem Minimum Mikrokontroler



Gambar 4.4. Rangkaian Sistem Minimum *Mikrokontroler* BAB V Timer,Counter, Interupsi dan real-time clock (RTC)

Mata Kuliah : **TEKNIK INTERFACE**

Capaian Pembelajaran :

Mahasiswa Mampu menerapkan dan mengembangkan teknik antarmuka, dalam sistem mikrokomputer sesuai dengan kebutuhan desain yang harus dipenuhi Kemampuan

Akhir yang Diharapkan :

Memahami Timer, Counter, Interupsi dan real-time clock (RTC)

Alokasi Waktu : 3 X 50 menit

Pertemuan ke : 5

Indikator :

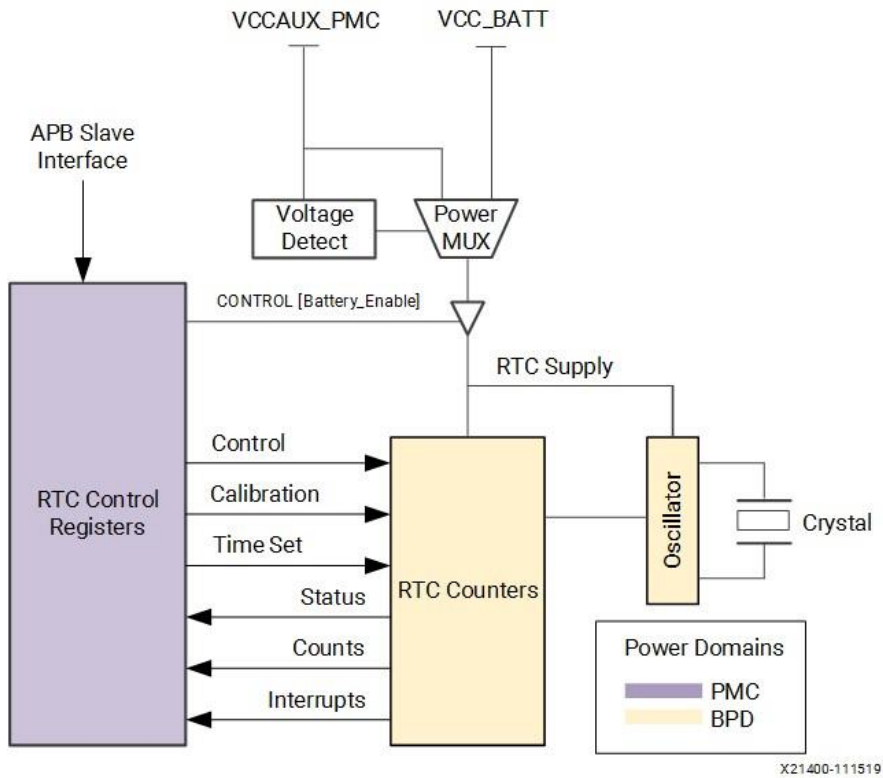
1. [C3] Mahasiswa akan mampu mengaplikasikan timer / counter ke mikrokomputer
2. [C3] Mahasiswa akan mampu mengaplikasikan interupsi internal dan eksternal ke mikrokomputer
3. [C3] Mahasiswa akan mampu mengaplikasikan watchdog ke mikrokomputer
4. [C3] Mahasiswa akan mampu mengaplikasikan RTC ke mikrokomputer

Real-Time Clock

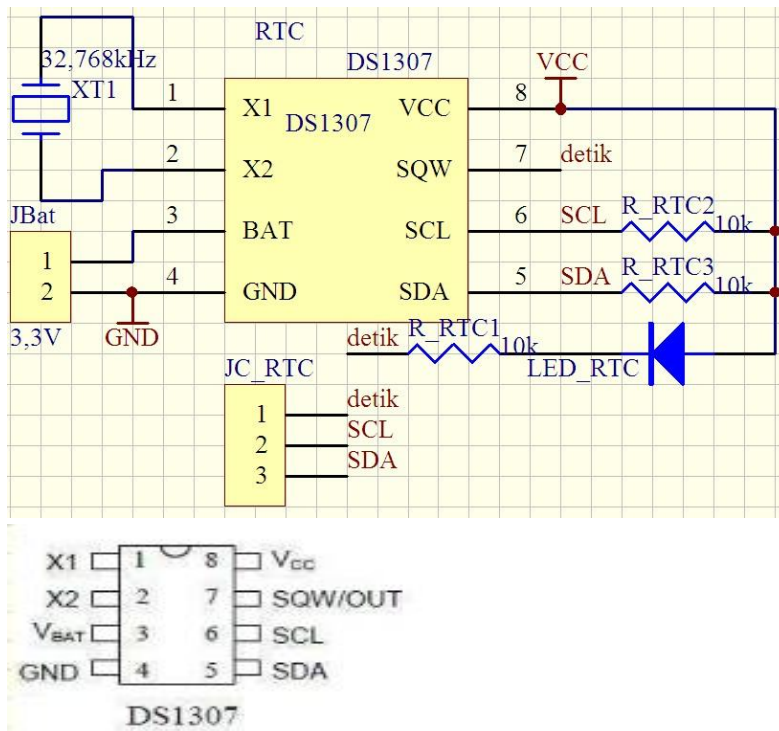
Jam waktu nyata (RTC) berada di PMC dan mempertahankan basis waktu yang akurat untuk perangkat lunak sistem dan aplikasi. Ini mencakup sirkuit kalibrasi untuk mengimbangi fluktuasi suhu dan tegangan dalam aplikasi yang membutuhkan akurasi yang lebih besar. RTC juga menyediakan pengaturan alarm dan fitur interupsi berkala. Jam real-time menyediakan operasi berkelanjutan yang didukung oleh suplai tambahan PMC (VCCAUX_PMC) atau suplai baterai (VCC_BATT). Ketika suplai tambahan tersedia, RTC menggunakannya untuk menjaga penghitung tetap aktif. RTC secara otomatis beralih ke catu daya baterai ketika pasokan tambahan tidak tersedia.

RTC menghasilkan dua sinyal interupsi sistem ke pengontrol interupsi generik (GIC), proksi GIC, dan logika yang dapat diprogram (PL) sekali setiap detik dan ketika peristiwa alarm terjadi. Interupsi tick kedua secara periodik dapat digunakan oleh semua prosesor sistem. Kontrol alarm harus dikelola pada tingkat sistem dengan prosesor.

Seperti yang ditunjukkan pada gambar berikut, subsistem RTC memiliki tiga modul utama: modul counter, modul register kontrol, dan modul osilator. Modul penghitung RTC ditenagai oleh domain daya baterai dan mencakup tiga penghitung, sirkuit kalibrasi, dan logika yang digunakan untuk mempertahankan waktu yang diprogram. Modul register kontrol RTC diimplementasikan dalam domain daya PMC dan menggabungkan semua register yang terkait dengan pengontrol RTC. Modul osilator dipasok oleh domain daya baterai dan menyediakan jam RTC.



Gambar 5.1. rangkaian RTC.



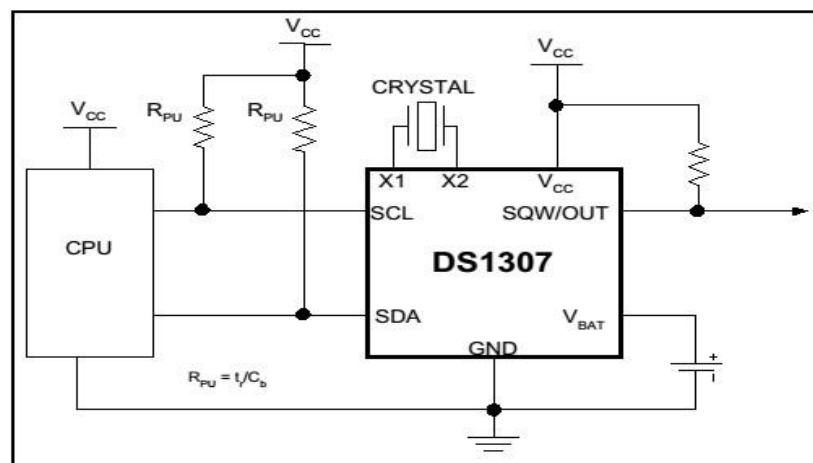
Gambar 5.2. IC DS1307

RTC ini kepanjangannya adalah Real Time Clock biasanya berupa IC yang mempunyai clock sumber sendiri dan internal baterai untuk menyimpan data waktu dan tanggal. Sehingga jika sistem jam digital dengan mikrokontroler tidak aktif, waktu dan tanggal didalam memori RTC tetap diupdate oleh sistem counter RTC. Salah satu RTC yang sudah populer dan mudah penggunaannya adalah DS1307.

Fitur-fitur DS1307:

- Real-time clock (RTC) menghitung detik, menit, jam, tanggal, bulan dan hari dan tahun valid sampai tahun 2100
- Ram 56-byte, nonvolatile untuk menyimpan data.
- 2 jalur serial interface (I2C)
- Keluaran gelombang kotak yg diprogram (SQW out)
- Automatic power-fail detect and switch
- Konsumsi arus hanya 500nA pada battery internal.
- mode dg oscillator running.
- temperature range: -40°C sampai +85°C

Untuk mengakses data tanggal dan waktu yg tersimpan di memori RTC DS1307 dapat dilakukan melalui komunikasi serial I2C



Gambar 5.3. rangkaian IC RTC DS1307

Koneksi ke Mikrokontroler.

RTC memiliki fitur-fitur berikut:

Pengoperasian terus menerus menggunakan catu daya tambahan atau baterai

Pengaturan alarm dan interupsi berkala

Sirkuit kalibrasi yang rumit untuk menjaga waktu yang sangat akurat

Penghitung detik 32-bit mewakili waktu 136 tahun

Tiga penghitung:

x penghitung detik 32-bit x penghitung centang 16-bit untuk mengukur detik

berdasarkan kristal 32 kHz x penghitung pecahan 4-bit untuk kalibrasi

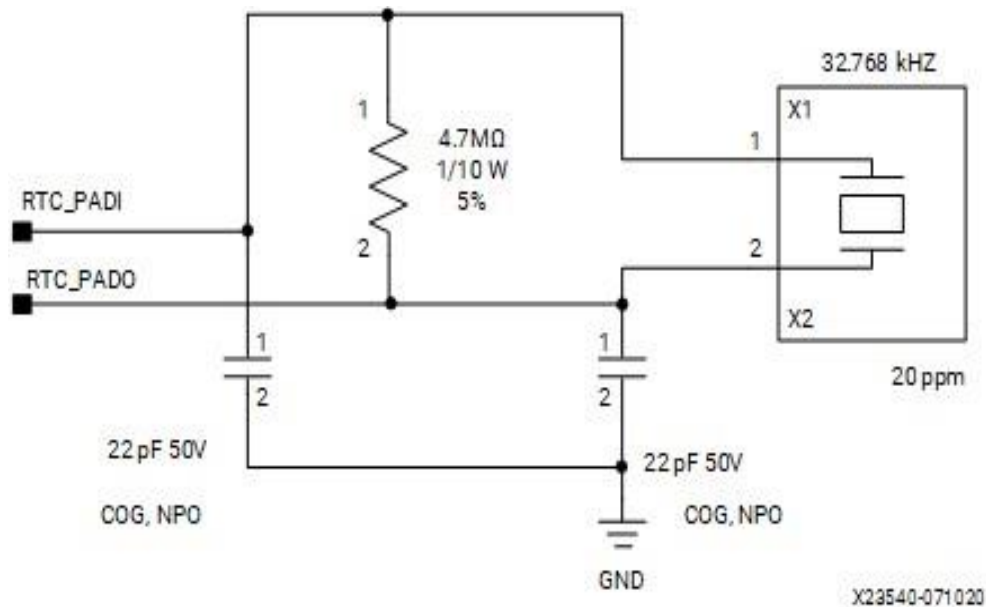
Akurasi RTC

Untuk osilator kristal 32,768 kHz, ketidakakuratan statis RTC dibatasi hingga $\pm 30,5$ ppm jika kristal yang dipilih memiliki ketidakakuratan statis yang lebih besar. Misalnya, ketidakakuratan kristal +50 ppm dalam satu juta tick menghasilkan 50 tick tambahan (atau turun $1-9/16$ tick setiap detik). Meningkatkan nilai kalibrasi dengan satu meninggalkan $9/16$ dari centang. Oleh karena itu, kristal statis +50 ppm dalam akurasi berdampak pada RTC yang serupa dengan kristal +17,17 ppm, karena beberapa ketidakakuratan diperhitungkan melalui kalibrasi detik.

Dengan mengaktifkan fitur kalibrasi fraksional, logika perhitungan kedua dapat melakukan kalibrasi lebih lanjut dengan menunda pembersihan penghitung tick sebesar satu hingga 15 tick osilator setiap 16 detik. Pada contoh sebelumnya, setelah setiap 16 detik, jam sembilan detik di depan. Oleh karena itu, dengan memprogram nilai sembilan ke dalam bidang kalibrasi fraksional dari register kalibrasi, waktu disesuaikan dengan sembilan tick setiap 16 siklus, yang mengoreksi ketidakakuratan statis osilator.

Kristal dan Sirkuit cloxk Eksternal

Kristal khas yang digunakan untuk RTC adalah kristal 20 ppm, 32,768 kHz (lihat gambar berikut). Menggunakan mekanisme kalibrasi RTC, ketidakakuratan efektif dikurangi menjadi kurang dari dua. Menggunakan kristal 65,536 kHz semakin mengurangi ketidakakuratan kalibrasi efektif hingga kurang dari 1 ppm.



Gambar 5.4. Sirkuit clock

Antarmuka dan Sinyal

Antarmuka RTC ke logika di PMC dan mencakup fitur-fitur ini: Antarmuka APB untuk mengakses register di dalam pengontrol dan penghitung RTCStatus interupsi, penyamaran interupsi, pengaktifan interupsi, dan register penonaktifan interupsi mengelola detik dan interupsi alarm

Register kontrol RTC mengaktifkan osilator kristal, mengontrol daya ke RTC, dan mengaktifkan kesalahan alamat ketika akses dibuat ke wilayah dalam ruang alamat RTC yang tidak dipetakan ke register

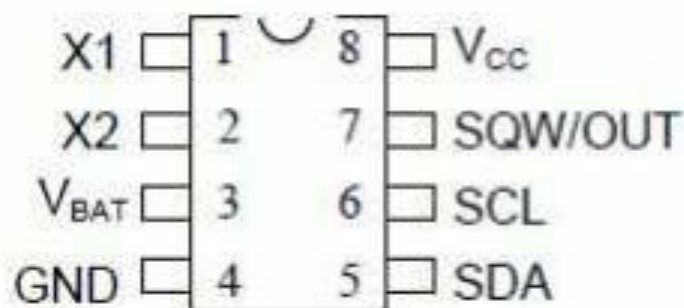
Interrupt dan Timer Control pada MCS-51

Interrupt MCS-51 8051 mempunyai 5 buah sumber interupsi. Dua buah interupsi eksternal, dua buah

interupsi timer dan sebuah interupsi port serial. Meskipun memerlukan pengertian yang lebih mendalam, pengetahuan mengenai interupsi sangat membantu mengatasi masalah pemrograman mikroprosesor/mikrokontroler dalam hal menangani banyak peralatan input/output. Pengetahuan mengenai interupsi tidak cukup hanya dibahas secara teori saja, diperlukan contoh program yang konkrit untuk memahami.

Saat kaki RESET pada IC mikroprosesor/mikrokontroler menerima sinyal reset (pada MCS51 sinyal tersebut berupa sinyal '1' sesaat, pada prosesor lain umumnya merupakan sinyal '0' sesaat), Program Counter diisi dengan sebuah nilai. Nilai tersebut dinamakan sebagai vector reset (reset vector), merupakan nomor awal memori-program yang menampung program yang harus dijalankan. Pembahasan di atas memberi gambaran bahwa proses reset merupakan peristiwa perangkat keras (sinyal reset diumpankan ke kaki Reset) yang dipakai untuk mengatur kerja dari perangkat lunak, yakni menentukan aliran program prosesor (mengisi Program Counter dengan vektor reset). Program yang dijalankan dengan cara reset, merupakan program utama bagi prosesor. Peristiwa perangkat keras yang dipakai untuk mengatur kerja dari perangkat lunak, tidak hanya terjadi pada proses reset, tapi terjadi pula dalam proses interupsi

RTC (Real Time Clock)



Gambar 5.5. Pin RTC DS1307 (Data Sheet IC)

RTC merupakan suatu chip (IC) yang memiliki fungsi sebagai penyimpan waktu dan tanggal. IC RTC yang digunakan dalam tugas akhir ini yaitu RTC DS1307 yang memiliki register yg dapat menyimpan data detik, menit, jam, tanggal, bulan dan tahun.

- Berikut ini keterangan dari fungsi masing-masing pin :
- VCC - Primary Power
 - Supply X1, X2 - 32.768kHz Crystal Connection
 - VBAT - +3V Battery Input
 - GND - Ground
 - SDA - Serial Data
 - SCL - Serial Clock
 - SQW/OUT - Square Wave/Output Driver

Timer Control MCS-51

Timer Control merupakan sarana input yang kurang dapat perhatian pemakai mikrokontroler, dengan sarana input ini mikrokontroler dengan mudah bisa dipakai untuk mengukur lebar pulsa, membangkitkan pulsa dengan lebar yang pasti, dipakai dalam pengendalian tegangan secara PWM (Pulse Width Modulation) dan sangat diperlukan untuk aplikasi remote control dengan infra merah. Pada dasarnya sarana input yang satu ini merupakan seperangkat pencacah biner (binary counter) yang terhubung langsung ke saluran-data mikrokontroler, sehingga mikrokontroler bisa membaca kedudukan pancacah, bila diperlukan mikrokontroler dapat pula merubah kedudukan pencacah tersebut. Seperti layaknya pencacah biner, bilamana sinyal denyut (clock) yang diumpankan sudah melebihi kapasitas pencacah, maka pada bagian akhir untaian pencacah akan timbul sinyal limpahan, sinyal ini merupakan suatu hal yang penting sekali dalam pemakaian pencacah. Terjadinya limpahan pencacah ini dicatat dalam sebuah flipflop tersendiri. Di samping itu, sinyal denyut yang diumpankan ke pencacah harus pula bisa dikendalikan dengan mudah. Hal-hal yang dibicarakan di atas diringkas dalam Gambar 4. MCS-51 mempunyai dua buah register timer/ counter 16 bit, yaitu Timer 0 dan Timer 1.

BAB VI

Antarmuka Digital Paralel

Mata Kuliah : TEKNIK INTERFACE

Capaian Pembelajaran :

Mahasiswa Mampu menerapkan dan mengembangkan teknik antarmuka, dalam sistem mikrokomputer sesuai dengan kebutuhan desain yang harus dipenuhi Kemampuan

Akhir yang Diharapkan :

Memahami Antarmuka Digital Paralel

Alokasi Waktu : 3 X 50 menit

Pertemuan ke : 2 Indikator :

- a. [C5] Mahasiswa akan mampu mendesain sistem mikrokomputer dengan peripheral masukan berupa keypad
- b. [C5] mahasiswa akan mampu mendesain mikrokomputer yang mempunyai peripheral keluaran berupa LED, 7-segmen, dan LCD alphanumerik

ANTARMUKA KOMUNIKASI PARALEL

Apa itu Komunikasi Paralel? Komunikasi paralel adalah metode pengiriman beberapa sinyal data secara bersamaan melalui link transmisi pada satu waktu. Ini terdiri dari beberapa saluran kabel secara paralel.

Dalam mode transfer paralel, ada transfer data satu kali dari sumber ke tujuan. Ini dimungkinkan menggunakan beberapa saluran untuk transfer data antara pemancar dan penerima

Perbedaan mendasar antara saluran paralel dan serial adalah jumlah berbagai kabel dalam bentuk fisik yang digunakan untuk transmisi data antara perangkat yang berbeda. Berbeda dengan transmisi serial, transfer data Paralel menggunakan lebih dari satu kabel (dan itu tidak termasuk kabel ground).

Mode komunikasi ini mahal karena memerlukan kabel dan perangkat keras tambahan. Namun demikian, itu dianggap sebagai mode transfer yang cepat. Printer, RAM, ISA, ATA, SCSI, PCI, Front side bus, dll., menggunakan transmisi data paralel. Mode Data Transfer



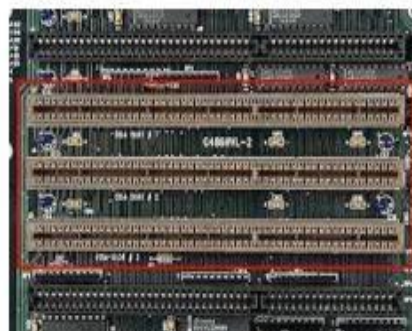
SCSI



PCI



ISA

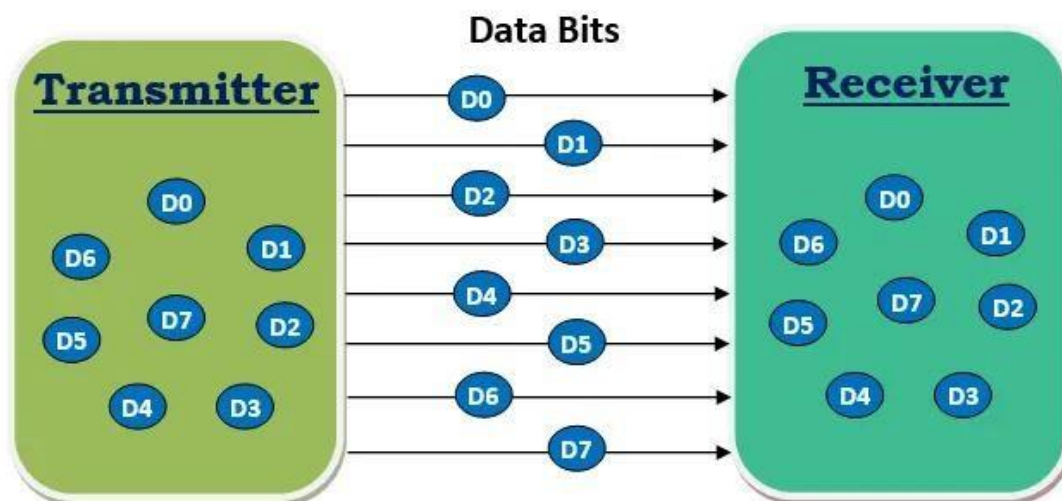


EISA

Gambar 6.1. Interface Hardware

Bagaimana Komunikasi Paralel Bekerja

Dalam komunikasi paralel, banyak bit (biasanya 8 bit atau kelipatannya) ditransfer pada saat yang sama di berbagai jalur paralel (kabel) dalam kabel yang sama dalam sinkronisasi dengan satu jam. Jam menyediakan waktu untuk transmisi sebagai sinyal clocking konstan melalui jalur paralel.



Gambar 6.1. Komunikasi Paralel

Komunikasi Paralel – Aliran Data melalui Jalur Paralel

Karena banyak bit ditransfer melalui berbagai jalur paralel pada saat yang sama, urutan string bit yang diterima mungkin berbeda atau tidak sinkron tergantung pada berbagai faktor seperti jarak sumber, lokasi, dan bandwidth yang tersedia. Contohnya adalah distorsi / skipping dalam panggilan internet (VOIP) dan streaming video.

Karakteristik Komunikasi Paralel

Sebelum pengembangan teknologi komunikasi serial kecepatan tinggi, pemilihan tautan paralel terhadap tautan komunikasi serial diputuskan oleh karakteristik berikut:

Transfer Data Berkecepatan Tinggi

Kecepatan pertukaran data dari data link komunikasi paralel setara dengan perkalian jumlah jalur paralel dan jumlah bit yang diproses pada satu unit waktu. Oleh karena itu, semakin banyak jalur paralel, semakin tinggi kecepatan transfer data yang dicapai.

Panjang Kabel Terbatas

Seiring bertambahnya panjang kabel, demikian juga jumlah kabel logam. Ini meningkatkan kemungkinan Cross-talk. Cross-talk berarti interferensi antara kabel-kabel dan itu mengarah pada pertukaran data yang dapat dibaca (nilai sampah) yang tidak berhasil.

Karena efek ini, kami tidak dapat menambah panjang kabel melebihi batas aman tertentu. Karena faktor ini, panjang kabel yang didukung oleh tautan paralel cukup pendek daripada tautan serial.

Instalasi Mudah

Tautan data paralel relatif mudah dipasang sebagai perangkat keras, yang pada gilirannya menjadikannya pilihan yang masuk akal. Konfigurasi port paralel di PC cukup mudah dibandingkan dengan rekan 'Serial link'. Hal ini karena hampir semua link serial memerlukan konversi ke dalam bentuk paralel menggunakan pemancar penerima asinkron Universal untuk memungkinkannya terhubung dengan bus data secara langsung.

Kapan Menggunakan Komunikasi Paralel Ini

harus digunakan ketika:

- Data Besar perlu ditransfer.

- Data yang akan dikirim sensitif terhadap waktu.

- Data diperlukan untuk ditransfer dengan cepat atau secara real time.

Perbedaan antara Komunikasi Serial dan Komunikasi Paralel
Perbedaan antara komunikasi Serial dan Paralel adalah sebagai berikut:
Tidak ada jalur untuk transfer Data:
Tautan serial hanya memiliki satu jalur untuk transfer data sementara yang Paralel memiliki banyak jalur.
Cakupan Jarak: Tautan serial mendukung komunikasi ke jarak jauh sementara transmisi data Paralel mendukung transmisi jarak terbatas.
Cross-talk: Peluang Cross-talk lebih sedikit di tautan Serial dibandingkan dengan Paralel karena hanya satu jalur yang tersedia.
Kecepatan: Tautan serial lambat dibandingkan dengan Tautan Paralel.
Komunikasi: Transmisi serial adalah dupleks penuh karena data dapat dikirim atau diterima di kedua ujungnya. Data paralel adalah setengah dupleks karena data dapat dikirim atau diterima.
Faktor Biaya: Tautan serial lebih murah daripada tautan Paralel karena lebih banyak jalur dalam mode paralel.
Kompleksitas: Tautan serial sederhana dan dapat diandalkan sementara tautan Paralel rumit dan tidak dapat diandalkan.

Terdapat 4 mode transfer data paralel yang dapat dilakukan di mana masing-masing mempunyai protokol tersendiri, yaitu:

1. Sederhana (simple)
2. Sederhana dengan strobe
3. Jabat tangan tunggal (single handshake)
4. Jabat tangan ganda (double handshake)

BAB VII

Antarmuka Serial

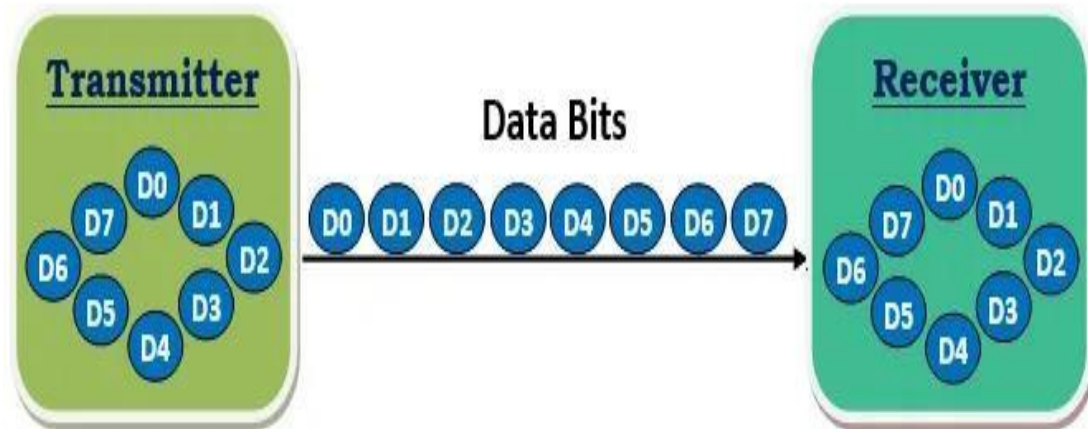
Mata Kuliah	: TEKNIK INTERFACE
Capaian Pembelajaran :	Mahasiswa Mampu menerapkan dan mengembangkan teknik antarmuka, dalam sistem mikrokomputer sesuai dengan kebutuhan desain yang harus dipenuhi Kemampuan
Akhir yang Diharapkan :	Mahasiswa mampu menerapkan Antarmuka Serial
Alokasi Waktu	: 3 X 50 menit
Pertemuan ke	: 2
Indikator	: <ol style="list-style-type: none">1. [C2] Mahasiswa akan mampu memahami prinsip kerja antarmuka Serial2. [C5] Mahasiswa akan mampu mendesain sistem mikrokomputer dengan antarmuka SPI3. [C5] Mahasiswa akan mampu mendesain sistem mikrokomputer dengan antarmuka I2C/TWI [C5] Mahasiswa akan mampu mendesain sistem mikrokomputer dengan antarmuka

Komunikasi Serial

Apa itu Komunikasi Serial Komunikasi serial dapat didefinisikan sebagai metode pengiriman satu bit data pada satu titik waktu secara berurutan melalui bus. Ambil contoh Busur anak panah. Bagaimana anak panah melesat dari busurnya? Satu per satu, bukan? Sama halnya dengan Komunikasi Serial.

Ini secara langsung berlawanan dengan fungsi komunikasi paralel, di mana beberapa bit dikirim bersama-sama pada bus yang terdiri dari banyak jalur kabel secara paralel. Komunikasi serial terjadi dengan hanya menggunakan satu kabel atau saluran. Oleh

karena itu, untuk komunikasi digital dua arah, kita hanya membutuhkan dua kabel antara pemancar dan penerima. Karena biaya untuk membangun komunikasi paralel sangat tinggi dan konfigurasinya panjang dan kompleks, komunikasi Serial lebih disukai untuk semua jaringan komunikasi/komputer utama.



Gambar 7.1. komunikasi Serial

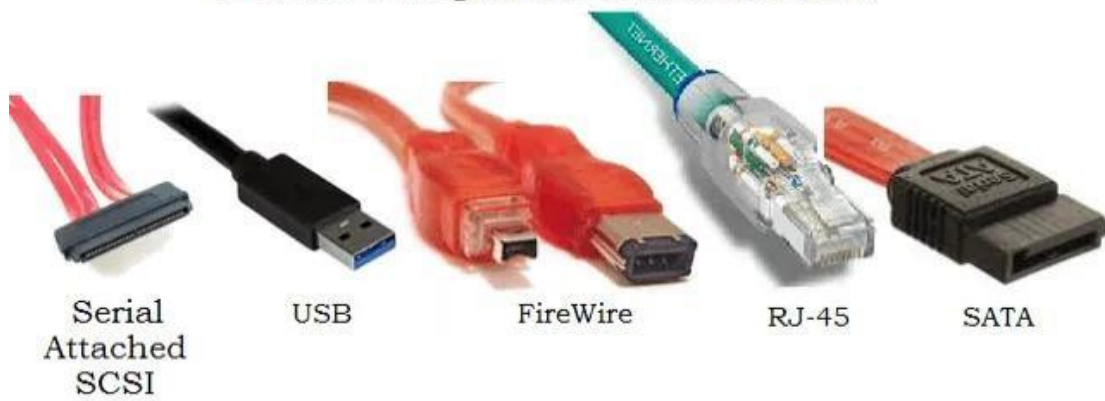
Komunikasi Serial – Aliran Data melalui Jalur Tunggal

Pada jarak yang kecil, bus serial menjadi lebih umum dan populer karena kelemahan dari bus paralel adalah keunggulan kesederhanaannya. Peningkatan teknologi untuk memastikan integritas sinyal serta meningkatkan kecepatan transmisi dan penerimaan data per saluran telah membuat port serial menjadi pesaing yang dekat dengan port paralel.

Biaya sirkuit terpadu banyak karena jumlah pin yang banyak. Oleh karena itu, banyak sirkuit terpadu menggunakan bus serial ketika pentingnya kecepatan tidak ada. Contoh bus serial berbiaya rendah ini adalah SPI, I²C, dan 1-Wire.

Beberapa contoh sistem yang terdiri dari komunikasi serial adalah Morse Code Telegraphy, RS-232, RS 422, RS-423, RS-485, Universal Serial Bus, FireWire, Serial Attached SCSI, Serial ATA, PCI Express, dll.

Devices Using Serial Communication



Gambar 7.2. device serial Perangkat

yang Menggunakan Komunikasi Serial

Protokol Komunikasi Serial

Berikut ini adalah Protokol Komunikasi Serial:

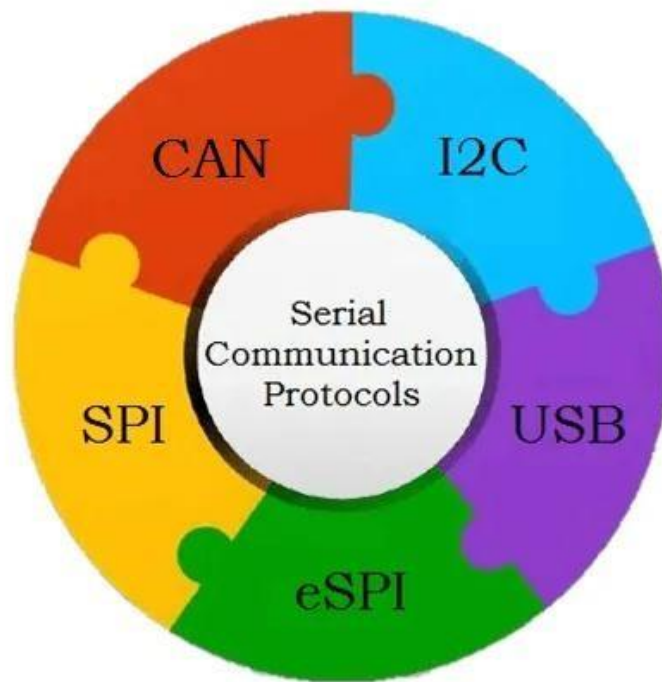
Protokol BISA

Protokol I2C

Protokol SPI

Protokol USB

Protokol eSPI



Gambar 7.3. Protokol Komunikasi Serial

Protokol BISA

Tahun 1970-an adalah era ketika produsen mobil mulai memperkenalkan fitur-fitur baru seperti pengereman anti-lock, AC, kontrol gigi, kunci pintu yang dioperasikan secara terpusat, dll. Fitur-fitur ini menjamin kabel ekstra besar dan desain kompleks yang meningkatkan biaya dan risiko. Untuk mengatasi masalah ini, Robert Bosch memperkenalkan protokol CAN pada tahun 1980-an. Protokol Komunikasi Serial ini selanjutnya distandarisasi sebagai ISO 11898 pada tahun 1993. Itu adalah protokol CAN yang sepenuhnya mengubah komunikasi antara sensor canggih. Protokol CAN digunakan untuk jaringan elektronik di mobil, pesawat terbang, dan sistem medis.

Protokol I2C

I2C adalah singkatan dari protokol Inter Integrated Circuits. Ini juga dikenal sebagai protokol IIC. Ini memungkinkan koneksi beberapa perangkat budak dengan satu atau lebih dari satu perangkat master. Komunikasi serial ini digunakan untuk komunikasi jarak pendek antara dua Sirkuit Terpadu pada basis yang sama (PCB). Ini menggunakan dua jalur transmisi dua arah untuk transfer data. Ini dapat mendukung hingga 3,4 Mbps yang merupakan kecepatan data yang sangat tinggi.

Protokol SPI

Protokol SPI adalah singkatan dari Serial Peripheral Interface Protocol. Itu diperkenalkan oleh Motorola pada 1980-an dan banyak digunakan dalam sistem tertanam. Ini digunakan untuk komunikasi jarak pendek dalam sistem tertanam. Arsitektur Master Slave digunakan saat komunikasi antar perangkat dalam protokol ini.

Protokol USB

Protokol USB tidak perlu diperkenalkan. Sejauh ini, ini adalah protokol paling populer yang digunakan. Protokol ini diperkenalkan untuk menstandarisasi koneksi periferal ke komputer. Dalam protokol USB, komunikasi data terjadi dalam bentuk paket 8 bit atau kelipatan 8 bit. Protokol eSPI eSPI Protocol adalah singkatan dari Enhanced Serial Peripheral Bus Interface Protocol.

Intel memperkenalkan eSPI Protocol sebagai pengganti bus Low Pin Count (LPC). Tujuan utamanya adalah untuk meminimalkan jumlah pin dibandingkan dengan bus LPC.

Standar Komunikasi Serial

Standar RS adalah standar paling populer yang didasarkan pada komunikasi serial.

Berikut ini adalah Standar RS utama yang banyak digunakan:

RS232

RS 422

RS 485

Standar RS dalam Komunikasi Serial



Gambar 7.4. Recommended standard dalam komunikasi serial

RS232

RS 232 adalah singkatan dari Recommended standard number 232. RS-232 , sebagai antarmuka, telah menjadi standar selama bertahun-tahun untuk interkoneksi Data Circuit Terminating Equipment (DCE) dengan Data Terminal Equipment (DTE). Berbagai bentuk antarmuka RS 232 tersedia di pasaran. Namun, karena berbeda dalam desain dan konfigurasi, mereka mudah dikonversi ke bentuk lain. Contoh antarmuka ini adalah RS 232 'C', RS 232 'D' , RS 232 V.10, RS 232 V.28.

RS 422

Standar ini diperkenalkan untuk memperbaiki kekurangan RS-232C. Kekurangan ini termasuk transmisi jarak jauh dan lambat. Waktu jam dan tujuannya ditentukan dalam standar ini. Contoh standar ini adalah konektor D Sub 9 pin dan D Sub 25 pin.

RS 485

Standar ini diperkenalkan untuk memperbaiki kekurangan RS 422. Kekurangan RS 422 adalah ketidakmampuan untuk mematikan sirkuit driver. RS 485 menggunakan tiga logika keadaan yang memungkinkannya untuk menonaktifkan setiap pemancar secara terpisah.

Secara lengkap perbedaan tiap parameter di antara RS 232, RS 422, dan RS 485 disajikan pada Tabel 7.1.

Tabel 7.1. Perbedaan di antara RS 232, RS 422, dan RS 485

PARAMETER	RS 232	RS 422	RS 485
Cabling	Single Ended	Differential	Differential
Number of Devices	1 Transmit and 1 Receiver	5 Transmitters and 10 Receivers	32 Transmitters and 32 Receivers
Communication Mode	Full Duplex	Full Duplex / Half Duplex	Half Duplex
Maximum Distance	50 Feet at 19.2 kbps	4000 feet at 100 kbps	4000 feet at 100 kbps
Maximum Data Rate	19.2 kbps at 50 Feet	10 Mbps at 50 feet	10 Mbps at 50 feet
Signaling Mode	Unbalanced	Balanced	Balanced
Mark (1)	-5 to -15 V	2V to 6V max. (B>A)	1.5V to 5V max. (B>A)
Space (0)	+5 to +15 V	2V to 6V max. (A>B)	1.5V to 5V max. (A>B)
Output current capability	500 mA	150 mA	250 mA

Perbandingan antara Standar RS Umum dalam Komunikasi Serial

Antarmuka Serial: RS-232

Transmisi data serial digunakan untuk komunikasi digital di antara

Sensor dan komputer

Komputer dan komputer

Komputer dan perangkat periferal (printer, stylus, mouse, ..)

Salah satu teknik komunikasi yang paling banyak digunakan untuk antarmuka peralatan eksternal

Protokol komunikasi serial: 1 bit pada satu waktu, berurutan

Transmisi paralel: 1 kata pada satu waktu (yaitu n bit secara paralel)

Keuntungan dari transmisi serial: kabel yang sangat sederhana Karakteristik

Transmisi RS-232

- jarak maksimum 20 meter
- Kecepatan bit maksimum 19 600 bps

Standar komunikasi serial alternatif:

- RS-422 (hingga 10 Mbps pada jarak 1,2 km)

- USB-1 & USB-2
- IEEE 1394 (Firewire / iLink) Karakteristik listrik
- Logika 1: -3V hingga -25V; biasanya -12V
- Logika 0: +3v hingga +25V; biasanya +12V
- Sinyal apa pun dalam kisaran -3V hingga +3V memiliki sinyal tak tentu keadaan logis
- Keadaan diam atau tidak aktif adalah -12V (yaitu logika 1)

Konektor

- DB25S adalah konektor 25 pin dengan fungsionalitas
- RS-232 penuh Soket komputer memiliki betina casing luar dengan penghubung pria pin
- Konektor kabel terminasi memiliki selubung luar pria dengan pin penghubung wanita

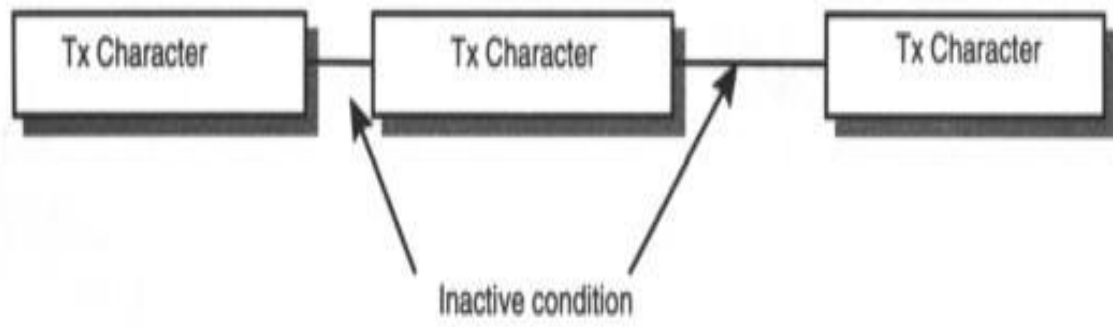


-
- Gambar 7.5 Pin Out DB 25

Format Data

Format data dalam pengiriman proses transmisi data pada RS-232 menggunakan komunikasi asinkron

Pola Format menggunakan Start-stop



Gambar 7.6 Pola transmisi data

BAB VIII

ADC dan DAC

Mata Kuliah : TEKNIK INTERFACE

Capaian Pembelajaran :

Mahasiswa Mampu menerapkan dan mengembangkan teknik antarmuka, dalam sistem mikrokomputer sesuai dengan kebutuhan desain yang harus dipenuhi Kemampuan

Akhir yang Diharapkan :

Memahami pangertian serta perancangan rangkaian ADC dan DAC

Alokasi Waktu : 3 X 50 menit

Pertemuan ke : 2 Indikator

:

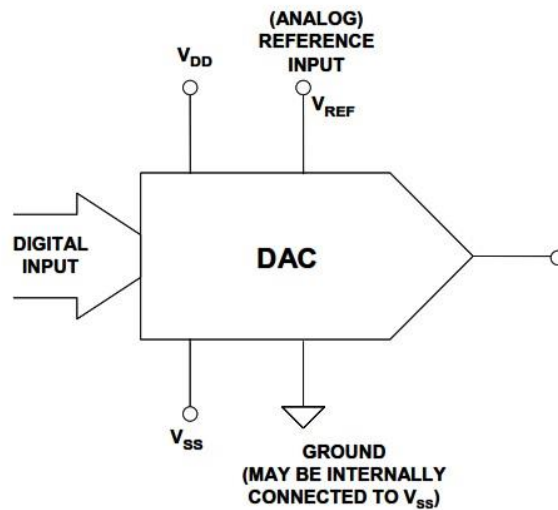
1. [C2] Mahasiswa akan mampu memahami prinsip ADC dan DAC
2. [C5] Mahasiswa akan mampu mendesain mikrokomputer dengan antarmuka ADC untuk peripheral sensor suhu dan kelembaban udara
[C5] Mahasiswa akan mampu mendesain mikrokomputer dengan antarmuka DAC untuk peripheral speaker DIGITAL ANALOG CONVERTER

ARCHITECTURES

Apa yang biasa kita sebut sebagai DAC hari ini biasanya sedikit lebih banyak. DAC akan biasanya memiliki konverter itu sendiri dan kumpulan sirkuit pendukung yang terpasang di dalam chip.

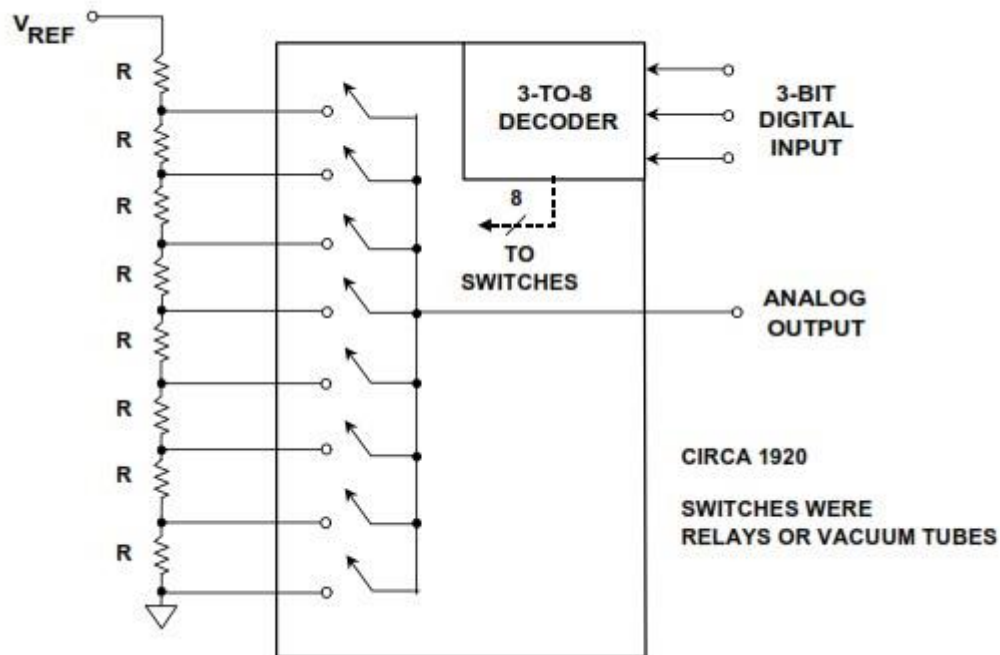
DAC pertama adalah desain tingkat papan, dibangun dari komponen terpisah, termasuk tabung vakum sebagai elemen switching. DAC monolitik mulai muncul di awal 70-an. Contoh-contoh awal ini sebenarnya adalah sub-blok DAC. Contoh dari ini akan menjadi AD550, yang merupakan sumber arus berbobot biner 4 bit. Arus ini blok sumber akan dikawinkan dengan bagian terpisah, seperti AD850, yang berisi a array resistor dan sakelar CMOS. Bersama-sama ini akan membentuk DAC dasar. Seperti yang kita pindah tepat waktu fungsi-fungsi ini terintegrasi pada cetakan yang sama, digital tambahan

sirkuit, khususnya kait untuk menyimpan input digital, ditambahkan. Kemudian peringkat kedua kait sering ditambahkan. Tujuan dari peringkat kedua adalah untuk memungkinkan mikroprosesor atau mikrokontroler untuk menulis ke banyak DAC dalam suatu sistem dan memperbarui semuanya di waktu yang sama. Peringkat input kait juga bisa menjadi register geser, yang memungkinkan antarmuka serial



Gambar 8.1. Kelvin Divider (String DAC)

Struktur paling sederhana dari semuanya adalah pembagi Kelvin atau string DAC seperti yang ditunjukkan pada Gambar 8.2. Versi N-bit dari DAC ini hanya terdiri dari 2^N resistor yang sama secara seri dan 2 switch (biasanya CMOS), satu di antara setiap node rantai dan output. Hasil diambil dari keran yang sesuai dengan menutup hanya salah satu sakelar (ada sedikit kompleksitas digital yang terlibat dalam decoding ke 1 dari 2^N beralih dari data N-bit).



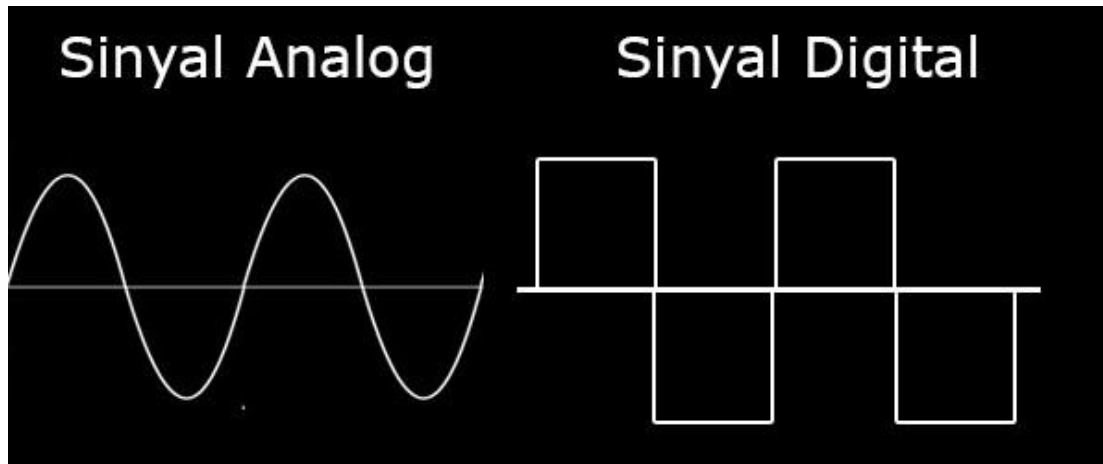
Gambar 8.2. Struktur pembagi Kelvin atau string DAC

Analog Digital Converter

adalah sistem yang mengubah sinyal analog, seperti suara yang ditangkap oleh mikrofon atau cahaya yang masuk ke kamera digital, menjadi sinyal digital. Sebuah ADC juga dapat memberikan pengukuran terisolasi seperti perangkat elektronik yang mengubah tegangan atau arus input analog ke angka digital yang mewakili besarnya tegangan atau arus. Biasanya keluaran digital adalah bilangan biner komplement dua yang sebanding dengan masukan, tetapi ada kemungkinan lain.

Ada beberapa arsitektur ADC. Karena kompleksitas dan kebutuhan akan komponen yang cocok secara tepat, semua kecuali ADC yang paling khusus diimplementasikan sebagai sirkuit terpadu (IC). Ini biasanya berbentuk chip sirkuit terpadu sinyal-logam-oksidasemikonduktor (MOS) yang mengintegrasikan sirkuit analog dan digital.

Konverter digital-ke-analog (DAC) melakukan fungsi sebaliknya; itu mengubah sinyal digital menjadi sinyal analog.



Gambar 8.3. Sinyal analog dan sinyal digital

BAB IX

Motor DC, Stepper dan Servo

Mata Kuliah : TEKNIK INTERFACE

Capaian Pembelajaran :

Mahasiswa Mampu menerapkan dan mengembangkan teknik antarmuka, dalam sistem mikrokomputer sesuai dengan kebutuhan desain yang harus dipenuhi Kemampuan

Akhir yang Diharapkan :

Memahami Motor DC, Stepper dan Servo

Alokasi Waktu : 3 X 50 menit

Pertemuan ke : 2

Indikator

- :
- a. [C3] Mahasiswa akan mampu mengaplikasikan antarmuka PWM untuk mengatur kecepatan motor DC
 - b. [C3] Mahasiswa akan mampu mengaplikasikan H-bridge untuk kendali putaran dan arah motor DC
 - c. [C5] Mahasiswa akan mampu mendesain sistem mikrokomputer untuk peripheral motor DC
 - d. [C5] Mahasiswa akan mampu mendesain sistem mikrokomputer untuk peripheral motor stepper
[C5] Mahasiswa akan mampu mendesain sistem mikrokomputer untuk peripheral motor servo

Interfacing Stepper motor / Control menggunakan 8085 and 8051

Stepper Motor

Motor stepper adalah perangkat yang menerjemahkan pulsa listrik menjadi gerakan mekanis dalam langkah dengan sudut langkah tetap.

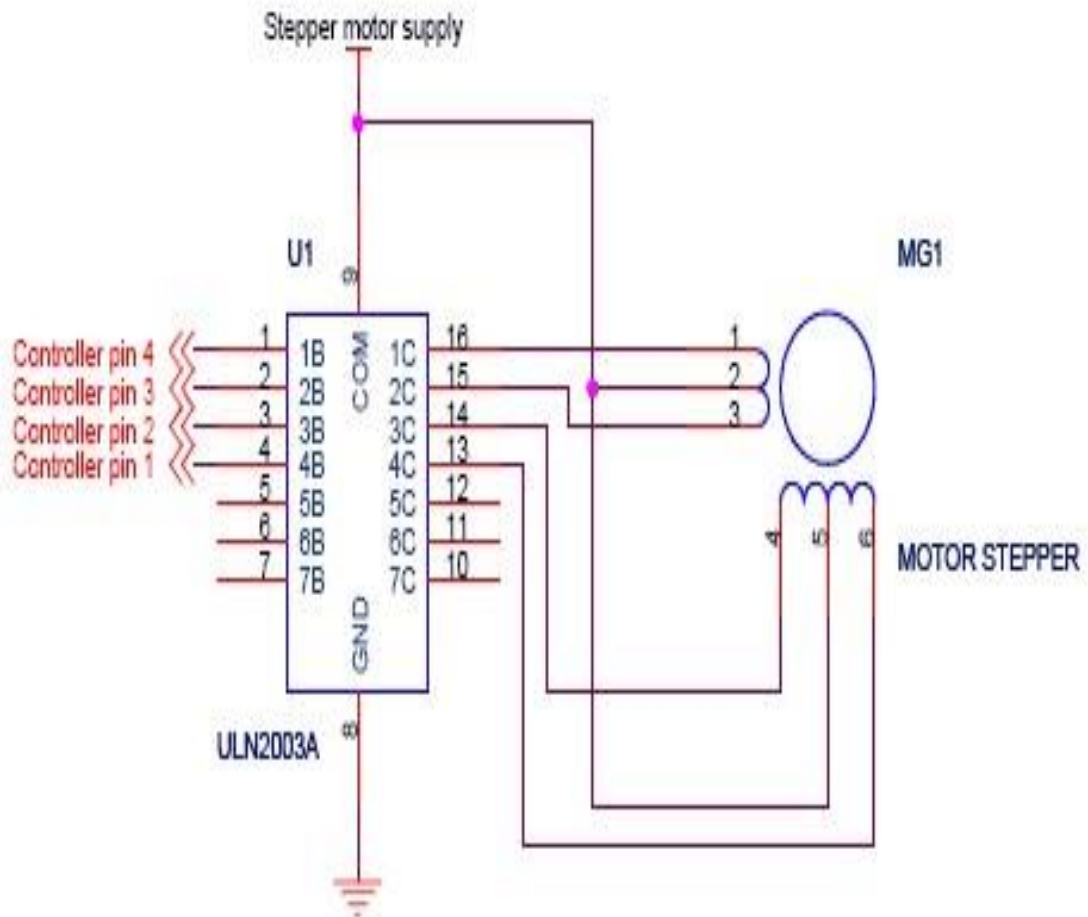
- Motor stepper berputar secara bertahap sebagai respons terhadap sinyal yang diberikan.
- Hal ini terutama digunakan untuk kontrol posisi.
- Digunakan dalam disk drive, printer dot matrix, plotter dan robotika dan sirkuit kontrol proses.

Struktur

Motor stepper memiliki magnet permanen yang disebut rotor (juga disebut poros) yang dikelilingi oleh stator. Motor stepper yang paling umum memiliki empat belitan stator yang dipasangkan dengan center-tap. Jenis motor stepper ini biasa disebut sebagai motor stepper empat fasa atau unipolar. Tap tengah memungkinkan perubahan arah arus di masing-masing dua kumparan ketika belitan diarde, sehingga menghasilkan perubahan polaritas stator. Antarmuka motor stepper kecil membutuhkan arus 400 mA untuk pengoperasiannya. Tetapi port mikrokontroler tidak dapat menghasilkan arus sebanyak ini. Jika motor tersebut terhubung langsung ke port mikroprosesor/mikrokontroler, motor

dapat menarik arus besar dari port dan merusaknya. Jadi sirkuit driver yang cocok digunakan dengan mikroprosesor/mikrokontroler untuk mengoperasikan motor.

Motor Driver Circuit (ULN2003)



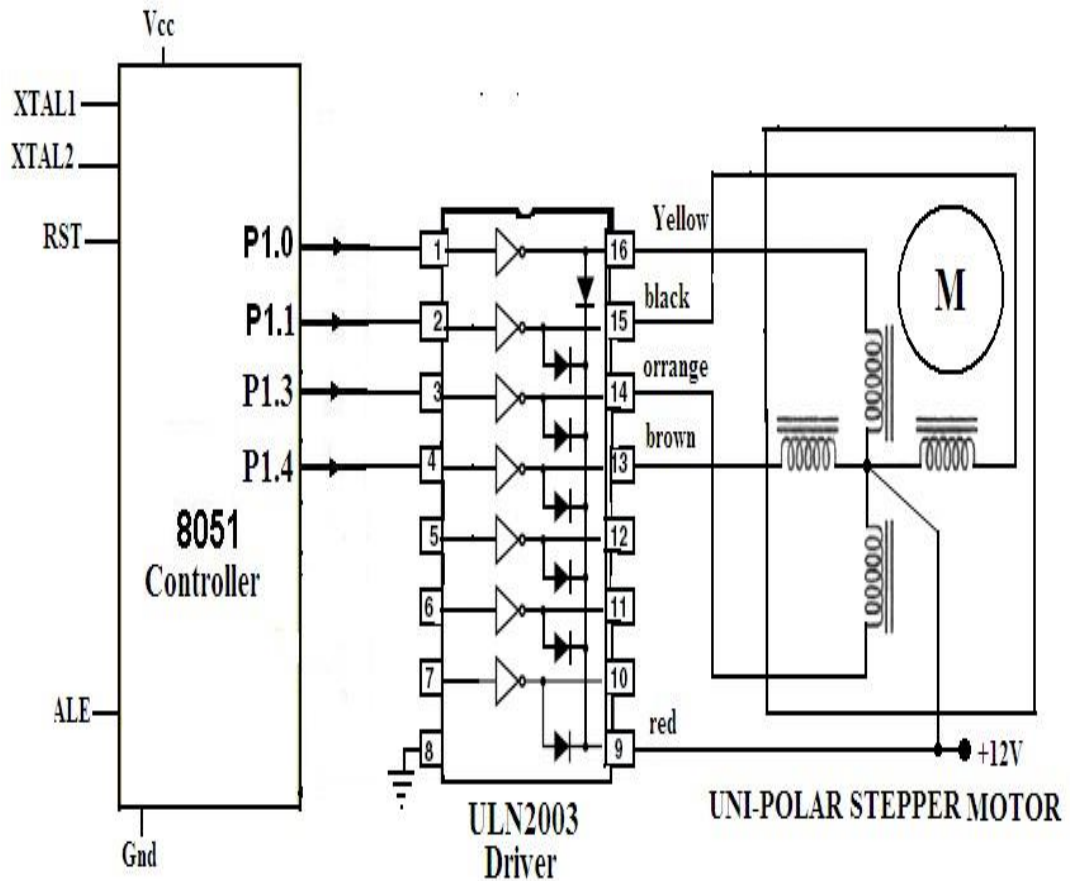
Gambar 9.1. Driver Motor stepper ULN2003

Antarmuka Bahkan motor stepper kecil membutuhkan arus 400 mA untuk pengoperasiannya. Tetapi port mikrokontroler tidak dapat menghasilkan arus sebanyak ini. Jika motor tersebut terhubung langsung ke port mikroprosesor/mikrokontroler, motor dapat menarik arus besar dari port dan merusaknya. Jadi sirkuit driver yang cocok digunakan dengan mikroprosesor/mikrokontroler untuk mengoperasikan motor. Sirkuit

driver motor stepper tersedia dalam bentuk IC. ULN2003 adalah salah satu IC driver yang merupakan rangkaian transistor Darlington Tegangan Tinggi Arus Tinggi dan dapat memberikan arus 500mA. Arus ini cukup untuk menggerakkan motor stepper kecil. Secara internal, ia memiliki dioda pelindung yang digunakan untuk melindungi motor dari kerusakan akibat ggl balik dan arus eddy yang besar. Jadi, ULN2003 ini digunakan sebagai driver untuk antarmuka motor stepper ke mikrokontroler.

Operasi Parameter penting dari motor stepper adalah sudut langkah. Ini adalah sudut minimum di mana motor berputar dalam menanggapi setiap pulsa eksitasi. Pada motor empat fasa jika terdapat 200 langkah dalam satu putaran penuh maka sudut langkah adalah $360/200 = 1,80$. Jadi untuk memutar motor stepper kita harus menerapkan pulsa eksitasi. Untuk ini pengontrol harus mengirim kode desimal heksa melalui salah satu portnya. Kode hex terutama tergantung pada konstruksi motor stepper. Jadi, semua motor stepper tidak memiliki kode Hex yang sama untuk putarannya. (lihat manual operasi yang disediakan oleh pabrikan.) Sebagai contoh, mari kita perhatikan kode hex untuk motor stepper untuk berputar searah jarum jam adalah: 77H , BBH , DDH dan EEH. Kode hex ini akan diterapkan ke terminal input driver melalui program bahasa assembly. Untuk memutar motor stepper berlawanan arah jarum jam, kode yang sama diterapkan dalam urutan terbalik

Stepper Motor interface- Schematic Diagram (for 8051)



Gambar 9.2. Circuit diagram Mikrokontroler dengan stepper

The assembly language program for 8051 is given below

ASSEMBLY LANGUAGE PROGRAM (8051)

```

Main : MOV A, # 0FF H      ; Initialization of Port 1 MOV
      P1, A              ;
      MOV A, #77 H       ; Code for the Phase 1
      MOV P1, A          ;
      ACALL DELAY        ; Delay
      subroutine MOV A, # BB H
      ; Code for the Phase II MOV P1, A
      ;
      ACALL DELAY        ; Delay subroutine.
      MOV A, # DD H      ; Code for the Phase III MOV P1, A
      ;
      ACALL DELAY        ; Delay subroutine

      MOV A, # EE H      ; Code for the Phase 1
      MOV P1, A          ;

```

ACALL DELAY ; Delay subroutine

SJMP MAIN; Keep the motor rotating continuously.

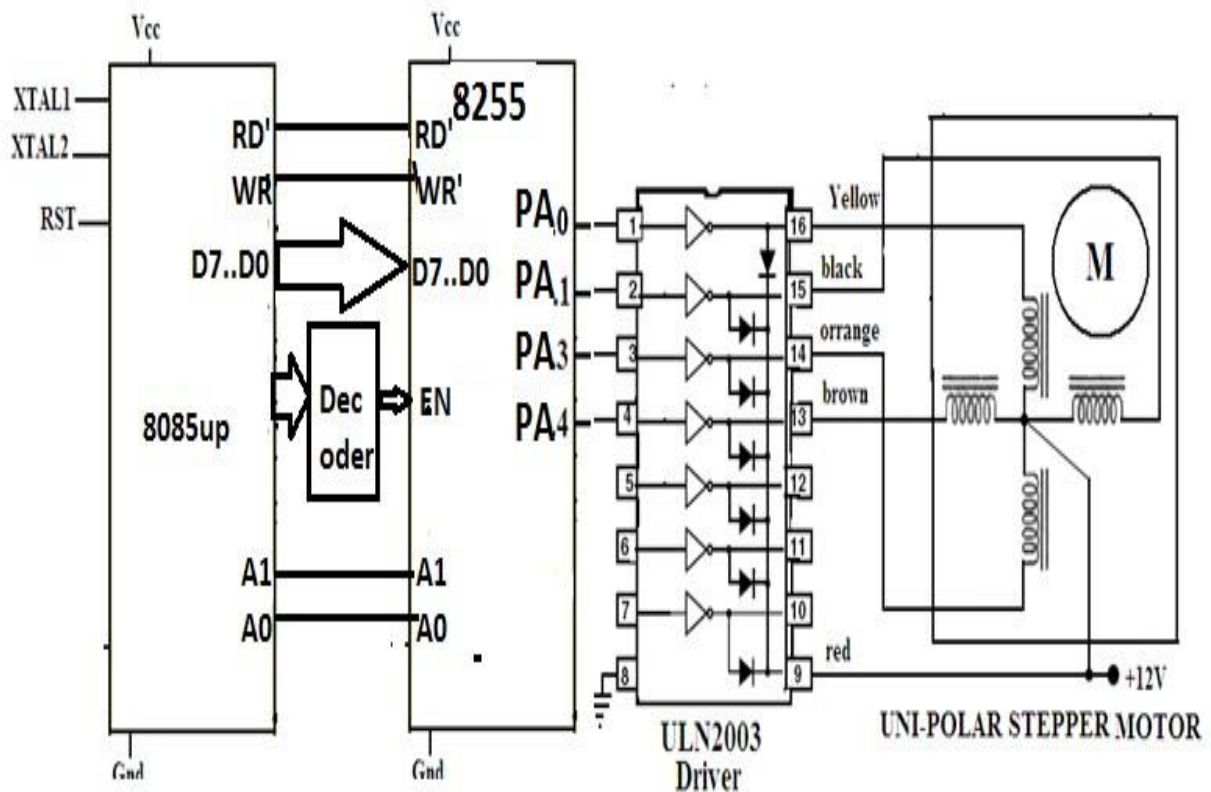
DELAY Subroutine

```

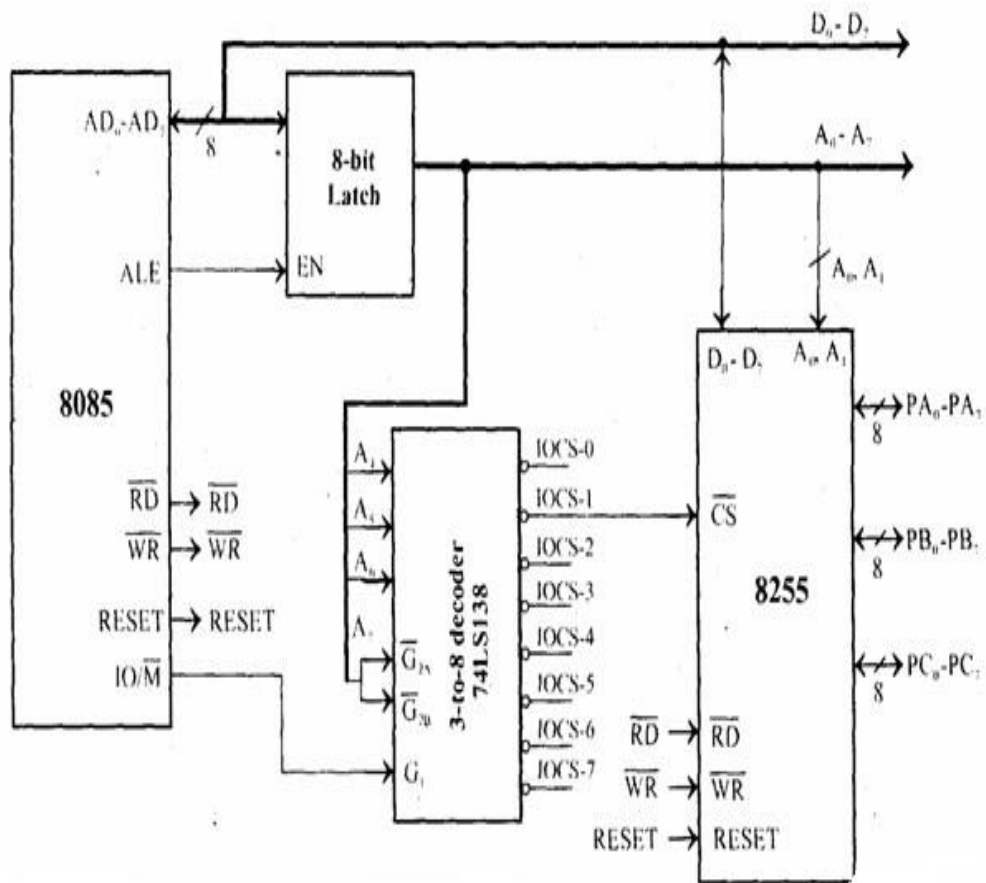
MOV R4, #0FF H ;
Load R4 with FF MOV R5, # 0FF
; Load R5 with FF
LOOP1: DJNZ R4, LOOP1 ; Decrement R4 until zero,wait
LOOP2: DJNZ R5, LOOP2 ; Decrement R5 until zero,wait
RET ; Return to main program .

```

Stepper Motor interface - Schematic Diagram for (8085)



Gambar 9.3. Diagram interface mikrokontroler



Gambar 9.5. detail diagram hubungan antara 8085 dan 82551

ASSEMBLY LANGUAGE PROGRAM (8085)

```

Main : MVI A, 80 ; 80H → Control word to configure
        PA,PB,PC in O/P OUT CWR_Address ; Write control word in
        CWR of 8255
        MVI A, 77 ; Code for the Phase 1

        OUT PortA_Address ; sent to motor via port A
        of 8255 ; CALL DELAY ; Delay
        subroutine
        MVI A, BB ; Code for the Phase II
        OUT PortA_Address ; sent to motor via port A of 8255

        CALL DELAY ;
        Delay subroutine. MVI A, DD
        ; Code for the Phase III

```

```

OUT PortA_Address      ; sent to motor via
port A of 8255; CALL DELAY      ;
Delay subroutine
MVI A, EE H           ; Code for the Phase 1

OUT PortA_Address      ; sent to motor via port A of 8255
; CALL DELAY           ; Delay subroutine
JMP MAIN              ; Keep the motor rotating continuously.

```

DELAY Subroutine

```

MVI C, FF             ; Load C with FF -- Change it for the speed
variation

LOOP1: MVI D,FF      ;
Load D with FF LOOP2: DCR D
JNZ LOOP2

DCR C

JNZ LOOP1

RET                  ; Return to main program .

```

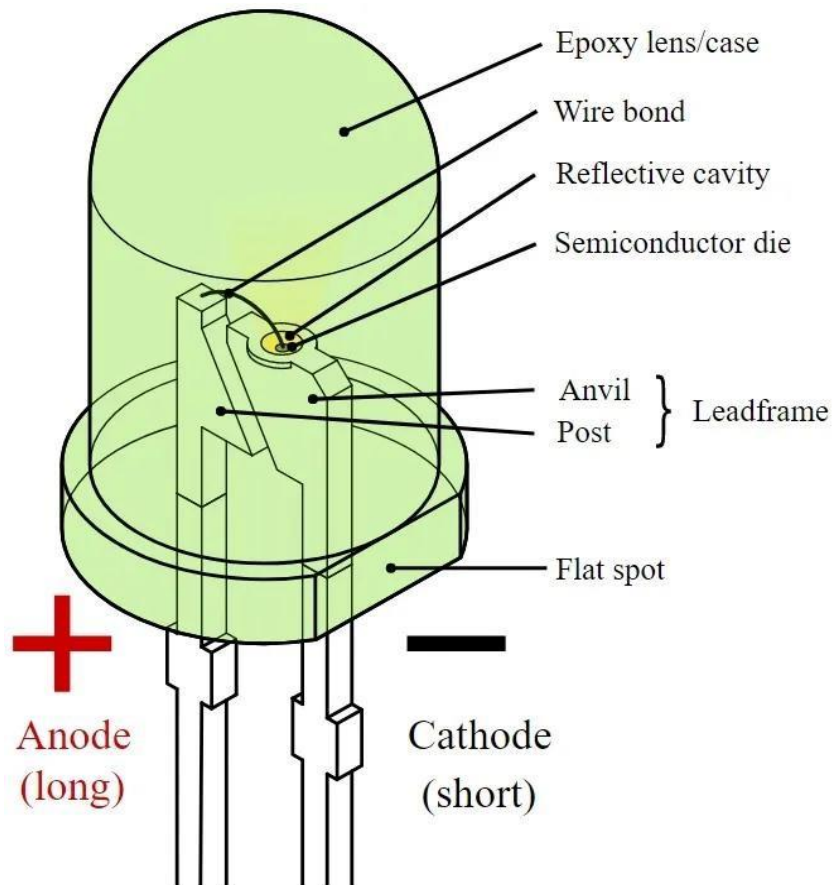
Antarmuka Dengan LED

Bagaimana cara kerja LED?

Light Emitting Diodes atau LED adalah komponen yang paling umum digunakan dalam banyak aplikasi. Mereka terbuat dari bahan semikonduktor. Prinsip kerja LED mirip dengan dioda, dimana LED menyala dalam kondisi bias maju dan mati dalam kondisi bias mundur. Anoda adalah terminal positif, dan katoda adalah terminal negatif dari LED. Ketika potensi yang lebih tinggi terhubung ke anoda dan potensi yang lebih rendah terhubung ke katoda, arus mengalir dari anoda ke katoda LED (potensi tinggi ke bawah), LED menyala dan dikatakan bias maju.

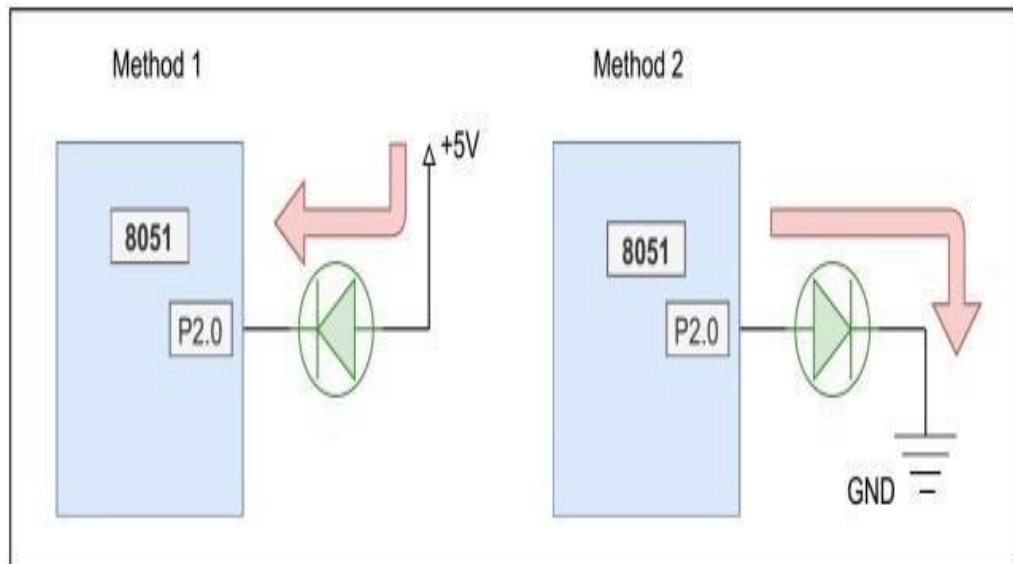


Gambar 9.5. LED RGB



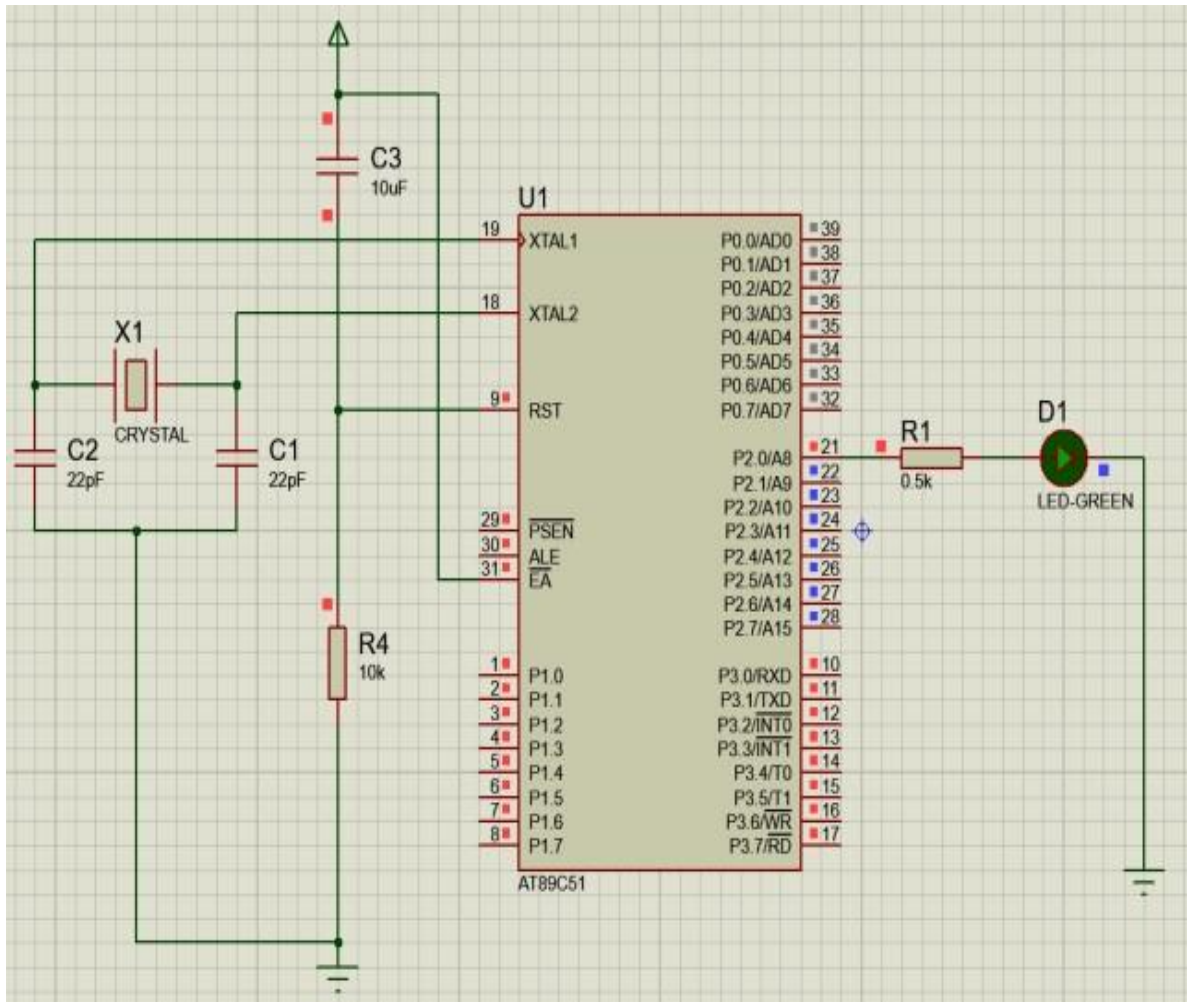
Gambar 9.6. struktur LED

Antarmuka LED langsung dengan 8051 Untuk menggunakan LED sebagai perangkat output, LED harus dihubungkan ke port mikrokontroler 8051 dan harus diprogram untuk berkedip



Gambar 9.7. interface LED dan Mikrokontroler

- Dalam metode 1, LED akan menyala hanya jika nilai PIN mikrokontroler 8051 LOW karena arus mengalir menuju PIN dari suplai +5V karena potensinya yang lebih rendah.
- Pada metode 2, LED hanya akan menyala jika nilai PIN mikrokontroler 8051 HIGH saat arus mengalir ke tanah.
- Dalam metode 1, LED akan menyala hanya jika nilai PIN mikrokontroler 8051 LOW karena arus mengalir menuju PIN dari suplai +5V karena potensinya yang lebih rendah.
- Pada metode 2, LED hanya akan menyala jika nilai PIN mikrokontroler 8051 HIGH saat arus mengalir ke tanah.



Gambar 9.8. Diagram sirkuit untuk langsung menghubungkan satu LED dengan 8051

Kami menghasilkan penundaan nominal 50 ms dengan menggunakan loop for bersarang. Anda dapat melihat tutorial tentang penghitung waktu dan penghitung 8051 untuk menghasilkan penundaan interval waktu yang diperlukan menggunakan penghitung waktu bawaan.

Di sini, kami menggunakan library reg51.h karena kami menggunakan mikrokontroler AT89C51/S51. Jika Anda menggunakan mikrokontroler AT89C52/S52, cukup ganti #include<reg.51> dengan

```
#include<reg.52> dan Anda siap melakukannya!#include<reg51.h> sbit
LED_pin = P2^0; // LED_pin is connected to P2.0
```

```
void ms_delay(unsigned int ms) // to provide a delay in ms
```

```

{   unsigned int i;
for(i=0;i<=ms;i++);
}

void main()
{
  P2 = 0x00; // to make P2 as output port
  while(1) // Repeat this loop until stopped
  {
    LED_pin = 1; // Turn ON the LED
ms_delay(50); // Delay of 50ms      LED_pin
= 0; // Turn OFF the LED
    ms_delay(50); // Delay of 50ms
  }
}

```

Indirect LED interfacing with 8051 using 8255 Programmable Peripheral Interface

Sebelum kita menggunakan Antarmuka Periferal yang Dapat Diprogram 8255, saya akan merekomendasikan Anda untuk memeriksa cara kerjanya, diagram pin, mode, cara memilih port (A, B, C, dan register kontrol) dan cara menyusun kata kontrol untuk kontrol register: 8255 Antarmuka Periferal yang Dapat Diprogram. Here, we'll use the Port A of 8255 in mode 0 by programming it using pin P1.0 and P1.1 of 8051.

Circuit Diagram

- Langkah 1: Jika Anda menggunakan Proteus atau dan perangkat lunak simulasi lainnya atau bahkan perangkat keras, pilih mikrokontroler AT89C51 atau AT89S51 atau varian lain yang kompatibel. (AT89C51 adalah mikrokontroler 8-bit dari keluarga Atmel yang bekerja dengan arsitektur 8051.)
- Langkah 2: Hubungkan osilator 12 MHz antara pin 18 dan 19.
- Langkah 3: Hubungkan dua kapasitor 22pF, dengan satu terminal di kedua sisi osilator dan terminal lainnya ke ground, seperti yang ditunjukkan di bawah ini.
- Langkah 4: Atur Pin 31, yaitu pin EA ke HIGH dengan menghubungkannya ke sumber DC +5V.

- Langkah 5: Sekarang, untuk membuat rangkaian RESET, sambungkan Pin 9 (RST) ke +5V melalui kapasitor 10 μ F dan sambungkan pin yang sama ke +0V (GND) melalui resistor 10k Ω atau potensiometer.
- Langkah 6: Hubungkan WR dan RD masing-masing 8255 PPI ke P3.6 dan P3.7, dari 8051. •
- Langkah 7: Hubungkan Port 2 (P2.0 – P2.7) masing-masing ke pin data (D0 – D7).
- **Step 8:** Connect CS, RESET, GND to ground, and VCC to +5V supply.
- **Step 9:** Connect A0 and A1 of 8255 PPI to P1.0 and P1.1, respectively, of 8051.
- **Step 10:** The final step is to connect PA0 – PA7 to cathode LEDs and anode to +5V VCC.

Koding C untuk antarmuka LED menggunakan 8255 PPI dengan 8051

Kita mulai dengan menjadikan Port 1, 2, dan 3 sebagai port output dengan mengirimkan 00H ke port tersebut. Kemudian kami menonaktifkan RD dan mengaktifkan WR dengan mengirimkan masing-masing 1 dan 0 karena mereka adalah pin rendah aktif (Menyala ketika pulsa rendah disediakan). Langkah selanjutnya adalah memilih mode 0 dari 8255 dengan mengirimkan pin 0 ke A1 dan A0. Sebelum kita mengirim pulsa untuk membuat LED menyala, kita harus menjadikan Port 8255 sebagai port output dengan mengirimkan 1000000 melalui port 2 ke 8255. Selanjutnya adalah logika pola di mana kita membuat LED berkedip. Pertama, kita akan mengedipkan LED yang terhubung ke nibble yang lebih rendah kemudian LED yang terhubung ke nibble yang lebih tinggi dengan penundaan di antaranya untuk memastikan "berkedip" terlihat.

```
#include<reg52.h>
sbit A0 = P1^0;
sbit A1 = P1^1;
sbit wr = P3^6;
sbit rd = P3^7;

void delay(unsigned int ms) // to provide a delay in ms
{
    unsigned int i;
    for(i=0;i<=ms;i++);
}
```

```

}

void main()
{
    P1 = 0x00; // Port-1 as Output
    P2 = 0x00; // Port-2 as Output
    P3 = 0x00; // Port-3 as Output
    delay (10000); // Delay before starting to
write   rd = 1; // Disabling read from   wr =
0; // Enabling write to   P2 = 0x80; // Making
all ports output
    A0 = 1; // Selecting control reg
    A1 = 1; // Selecting control reg

    while(1)
    {
        wr
= 1;
        A0 = 0; // Select Port A as output
        A1 = 0;   delay (10000); // Sufficient delay is to be
provided to see the blinking
        P2 = 0x0F; // To make LEDs connected to higher nibble glow as we're
using method 1
        wr = 0;
        // You can also write P2 = ~0x0F to make LEDs connected
to lower nibbles light up.   wr = 1;
        A0 = 0; // Select Port A as output
        A1 = 0;
        delay (10000); // Sufficient delay is to be provided to see the blinking
        P2 = 0xF0; // To make LEDs lower nibble light up as we're using
method 1
        wr = 0;
    }
}

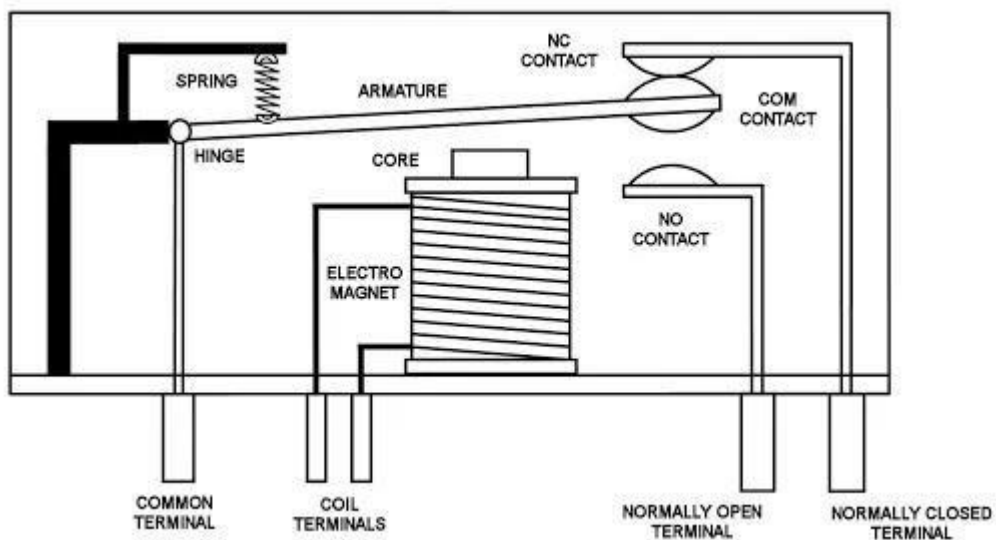
```

Menghubungkan 8051 dengan relai untuk menggerakkan periferal berdaya tinggi katakanlah Anda ingin mengontrol peralatan di rumah Anda menggunakan mikrokontroler. Tapi inilah masalahnya. Dalam hal daya, mikrokontroler tidak dapat mengeluarkan watt tinggi. Itu mungkin membuat Anda berpikir bahwa perangkat berdaya rendah seperti mikrokontroler mungkin tidak dapat mengontrol periferal berdaya tinggi. Di sinilah peran relai. Singkatnya, relai adalah sakelar yang memungkinkan mikrokontroler mengontrol perangkat eksternal berdaya tinggi. Ini adalah gerbang yang menerima sinyal kontrol dari mikrokontroler, tetapi sumber sinyal daya dari sumber daya tinggi. Ini semacam membuat dua sinyal bekerja bersama-sama untuk memungkinkan pengontrol mengontrol peralatan bertenaga

tinggi. Dalam posting ini, kita akan memahami dasar-dasar relay dan jenisnya. Dan kami akan menghubungkan relay ini dengan mikrokontroler 8051.

Apa itu Relay?

Relai adalah sakelar yang dioperasikan secara elektrik yang menggunakan rangsangan elektronik kecil dari perangkat eksternal untuk mengontrol periferil berdaya tinggi. Konstruksi relay membantu menyediakan isolasi listrik antara dua sirkuit yang sangat penting dalam hal mikrokontroler. Karena sifat isolasi listrik dan daya dukung beban yang tinggi dari relay, relay digunakan untuk mengontrol perangkat eksternal berdaya tinggi seperti kipas, lampu, dan motor. Ada dua jenis relay: 1. Relai elektromekanis 2. Relai keadaan padat. Relai Elektromekanis Seperti namanya, relai elektromekanis menggunakan elektromagnet dan kontak mekanis untuk mengontrol perangkat. Ia menggunakan prinsip elektromagnetisme untuk menggerakkan anker mekanik. Mari kita lihat bagian dalam relai elektromekanis untuk mendapatkan



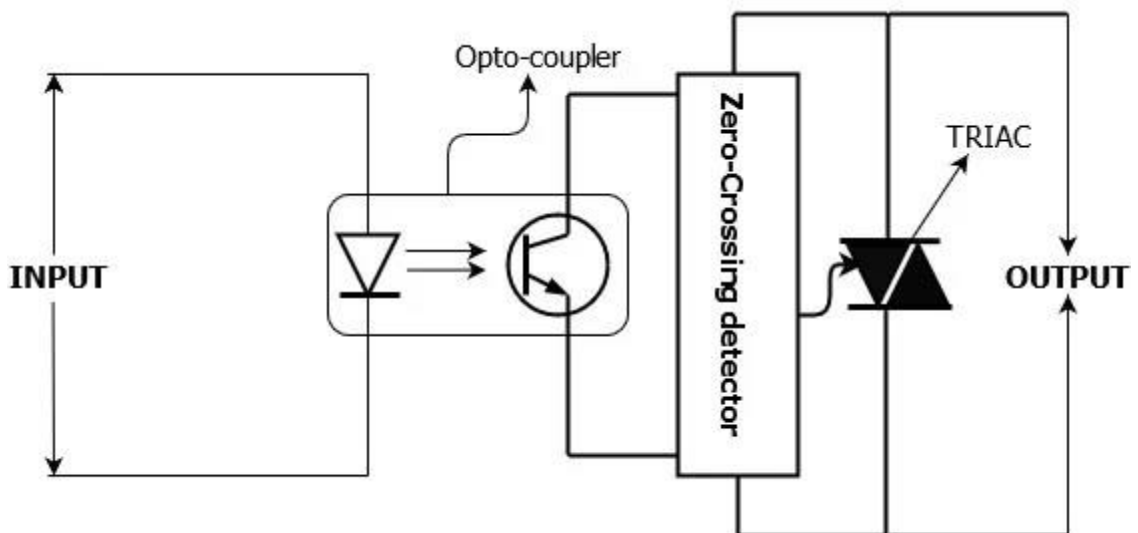
Gambar 9.9. Struktur Relay

Relai elektromekanis terdiri dari elektromagnet, yang, ketika diberi energi, menggerakkan anker logam. Armature terhubung ke kontak yang biasanya

tertutup ketika elektromagnet tidak diberi energi. Setelah diberi energi, elektromagnet menarik angker ke kontak yang biasanya terbuka yang melengkapi rangkaian yang menggerakkan perangkat eksternal.

Solid-state relays

Relay keadaan padat Relai solid-state, berbeda dengan relai elektromekanis, tidak memiliki bagian yang dapat bergerak. Karena alasan ini, relai solid-state memiliki waktu switching yang lebih cepat dan siklus hidup yang lebih tinggi jika dibandingkan dengan relai elektromagnetik. Relay solid-state menggunakan optocoupler alih-alih elektromagnet untuk menyediakan isolasi dan kontrol listrik. Dalam kasus perangkat output DC, relai solid-state menggunakan penyearah yang dikontrol silikon dengan optocoupler sedangkan, dalam kasus beban AC, mereka menggunakan TRIAC atau beberapa



Gambar 9.10. rangkaian Solid-state relays

Gambar di atas menggambarkan cara kerja relai solid-state dengan beban AC menggunakan TRIAC dan detektor zero-crossing. Relai keadaan padat vs Relai elektromekanis disajikan pada Tabel 9.1.

Tabel 9.1. Relai keadaan padat vs Relai elektromekanis

Solid-state relays	Electromechanical relays
Karena penggunaan detektor zero-crossing, relai solid-state meminimalkan kebisingan listrik.	Relai elektromekanis menggunakan kumparan yang bertindak sebagai induktor, dan dapat menyebabkan gangguan listrik seperti backEMF .
Memiliki siklus hidup yang lebih lama karena memiliki lebih sedikit bagian yang bergerak.	Karena banyak bagian yang bergerak, mereka memiliki siklus hidup yang lebih pendek.
Memiliki waktu respons yang lebih cepat dan konsumsi daya yang lebih rendah.	Memiliki waktu respons yang lebih lambat karena sakelar mekanis mengontrol perangkat eksternal.
Gunakan optocoupler untuk mekanisme pengontrolan.	Gunakan elektromagnet untuk mekanisme pengontrolan.
Sangat tahan terhadap guncangan dan getaran karena tidak memiliki banyak bagian yang bergerak	Tidak tahan terhadap getaran karena memiliki banyak bagian yang bergerak.
Dapat digunakan di tempat-tempat dengan gaya elektromagnetik yang tinggi.	Tidak dapat digunakan di tempat dengan nilai gaya elektromagnetik yang tinggi.
Memiliki biaya awal yang lebih tinggi.	Lebih murah dibandingkan dengan solid-state relay.
Karena mereka menggunakan semikonduktor untuk mengontrol sirkuit keluaran, lonjakan tegangan dapat menyebabkan kerusakan pada mereka .	Variasi eksternal dari sisi output tidak merusak perangkat karena menggunakan sakelar fisik.

Mengapa menggunakan Relay?

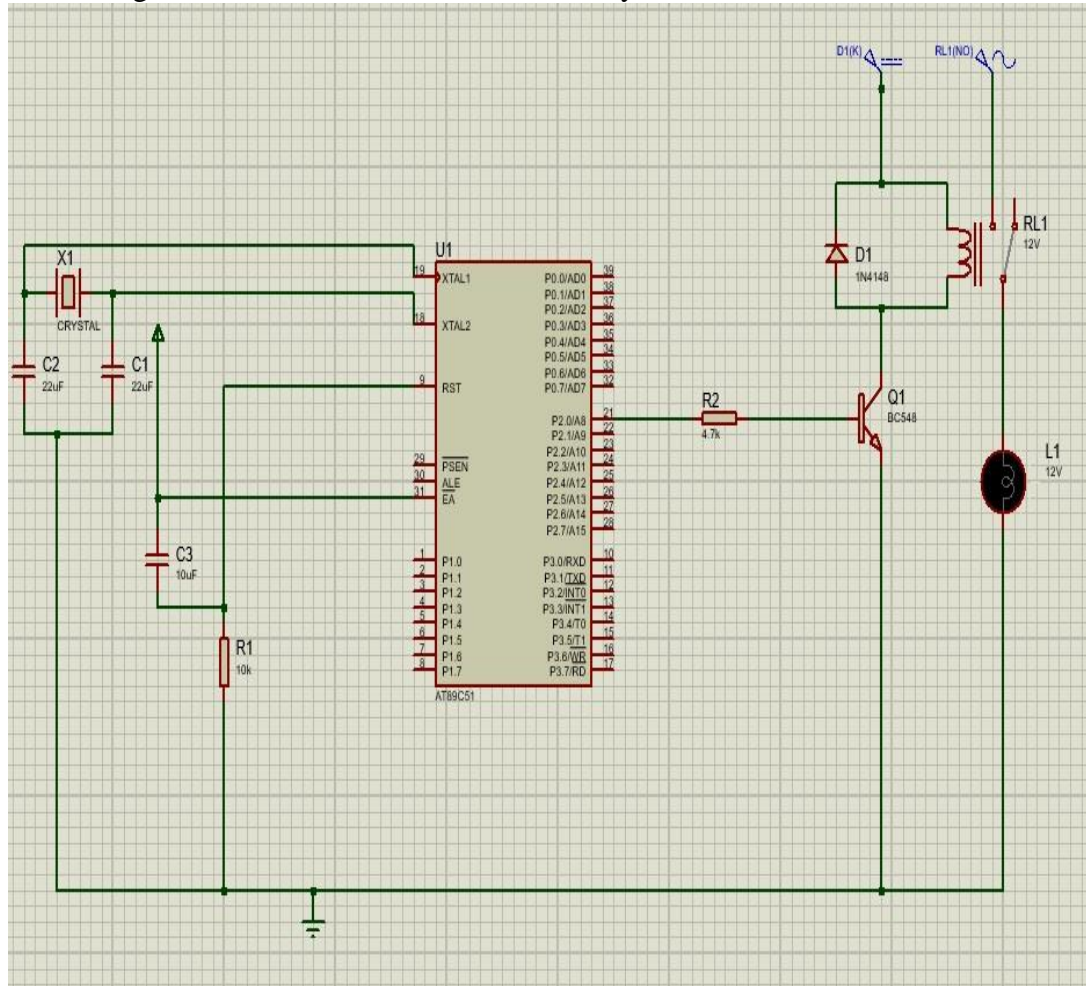
Ketika datang untuk mengendalikan perangkat dengan kebutuhan daya tinggi, mikrokontroler sama sekali tidak cocok untuk tugas sumber daya. Relay bertindak sebagai perangkat switching yang menggunakan sinyal kontrol daya rendah dari mikrokontroler untuk mengontrol perangkat eksternal dengan menggunakan elektromagnet/optocoupler. Mereka menyediakan cara bagi mikrokontroler untuk mengontrol periferan daya tinggi sambil memberikan isolasi listrik ke mikrokontroler. Menghubungkan mikrokontroler langsung ke relai bukanlah cara yang tepat. Alasan utamanya adalah kenyataan bahwa arus dari mikrokontroler terlalu kecil untuk mengontrol mekanisme pensaklaran relai. Karena itu, transistor diperlukan untuk memperkuat sinyal. Selain itu, dalam kasus relai elektromagnetik, dioda diperlukan untuk mencegah EMF belakang merusak transistor dan mikrokontroler.

Komponen yang diperlukan untuk interfacing relay dengan 8051 1.

Mikrokontroler 8051 – AT89C51/AT89S51/52 atau varian serupa.

2. Relai elektromekanis 12 v
3. Relai keadaan padat
4. Transistor BC 548 NPN
5. Dioda roda gila 1N4148
6. Kristal osilator – 12Mhz
7. Kapasitor – 22pF x 2, 10 μ F x 1
8. Resistor – 10k Ω x 1, 4.7k Ω x 2
9. Pasokan 12V DC

Interfacing 8051 with an electromechanical relay



Gambar 9.11. rangkaian interface untuk mikrokontoler dengan relay

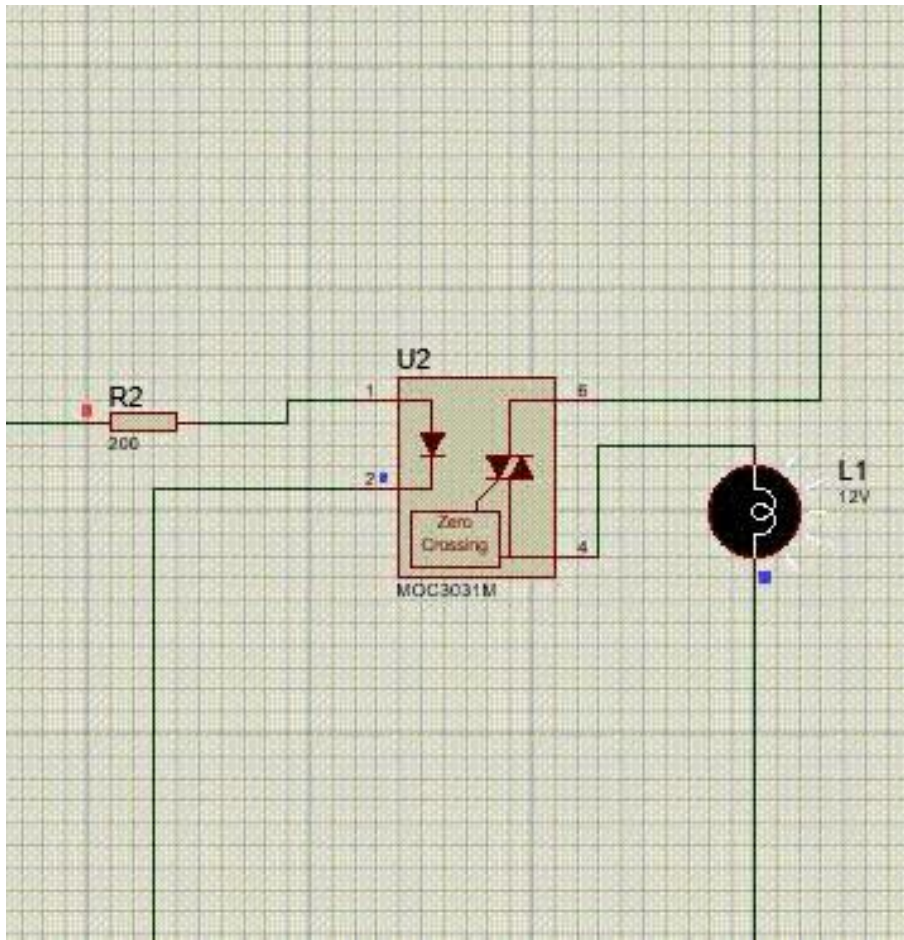
- Untuk menghubungkan relai dengan 8051, ikuti langkah-langkah berikut. Dalam kasus kami, kami menggunakan Proteus sebagai perangkat lunak simulasi dan varian AT89C51 dari mikrokontroler 8051.
- Hubungkan rangkaian osilator ke pin 19 dan 20. Ini termasuk osilator kristal dan dua kapasitor masing-masing 22µF. Hubungkan mereka ke pin, seperti yang ditunjukkan pada diagram.
- Hubungkan salah satu ujung kapasitor ke pin EA dan ujung lainnya ke resistor. Hubungkan resistor ini ke pin RST, seperti yang ditunjukkan pada diagram.
- Kami menggunakan port 2.0 sebagai port output, untuk mengontrol relay menggunakan rangkaian driver relay.
- Hubungkan salah satu ujung port 2.0 ke resistor 4.7k ohm. Hubungkan ujung resistor yang lain ke dasar transistor. Ini menyalakan transistor ketika arus kecil mengalir melalui yang sama — memungkinkan arus mengalir melaluinya.
- Ground kolektor transistor.
- Hubungkan emitor ke ujung positif dioda untuk memblokir EMF belakang dari relai. Hubungkan ujung lainnya ke salah satu terminal relai.

- Hubungkan ujung negatif dioda ke catu daya 12V DC dan ujung lainnya ke daya relai.
- Hubungkan salah satu ujung terminal relai ke sumber AC 220v dan 60Hz. Hubungkan ujung lainnya ke lampu atau periferal lain yang ingin Anda kendalikan menggunakan relai.
- Mengardekan terminal periferal lainnya.

C code to interface a relay with 8051

```
#include<reg52.h> sbit relay_pin = P2^0; gives a
variable name to port 2 void Delay_ms(int);
predefines the delay function void main()
{
do // loop to contineously switch the relay
{ relay_pin = 0;
Delay_ms(1000);
relay_pin = 1;
Delay_ms(1000);
}while(1); }
void Delay_ms(int k) // function to generate delay
{
int j; int
i;
for(i=0;i<k;i++)
{ for(j=0;j<1275;j++)
}
}
```

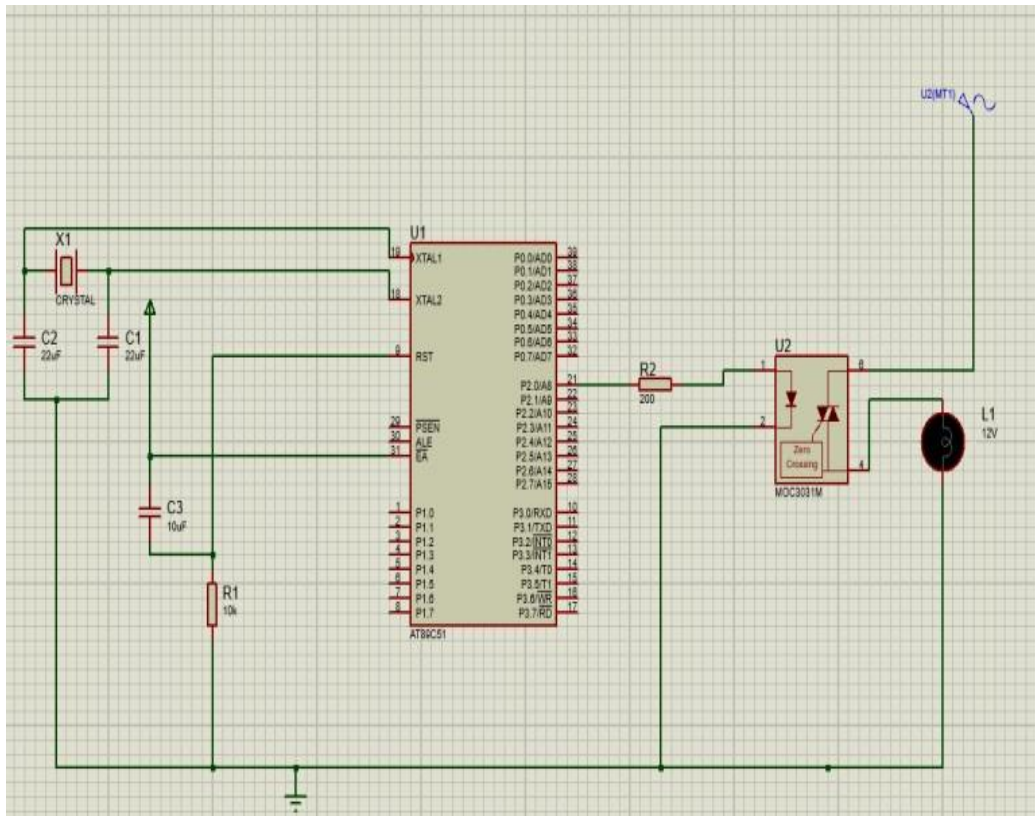
Kode yang diberikan di atas digunakan untuk menyalakan dan mematikan lampu eksternal yang terhubung ke relai setiap 1000 milidetik. Untuk melakukan ini, mikrokontroler mengirimkan pulsa ke port 2.0 di mana rangkaian penggerak relai terhubung. Mikrokontroler mengirimkan sinyal tinggi ke dasar transistor, yang memungkinkan relai untuk diisi. Dioda memberikan perlindungan kepada transistor dari EMF belakang induktor di elektromagnet.



Gambar 9.12. diagram solid state relay

Interfacing 8051 dengan solid-state relays

Menghubungkan relai keadaan padat ke 8051 mirip dengan relai elektromekanis, tetapi tidak memerlukan rangkaian driver. Alasan utamanya adalah karena tidak menimbulkan masalah seperti back EMF dan dapat dikontrol langsung dengan menggunakan mikrokontroler



Gambar 9.13 Rangkaian solid state relay dengan mikrokontroler

Kode yang sama yang diberikan di atas dapat digunakan untuk mengontrol relai solid-state juga. Kami harap membaca artikel ini telah membantu Anda memahami relai dan cara menghubungkannya dengan mikrokontroler dengan cara yang lebih baik. Jika Anda memiliki masalah dalam memahami hal yang sama, jangan ragu untuk menghubungi kami melalui bagian komentar dan kami akan segera menghubungi Anda kembali. Pada posting berikutnya dalam kursus 8051 ini, kita akan menghubungkan modul Bluetooth ke 8051 dan menggunakannya untuk mengontrol alat melalui relai.

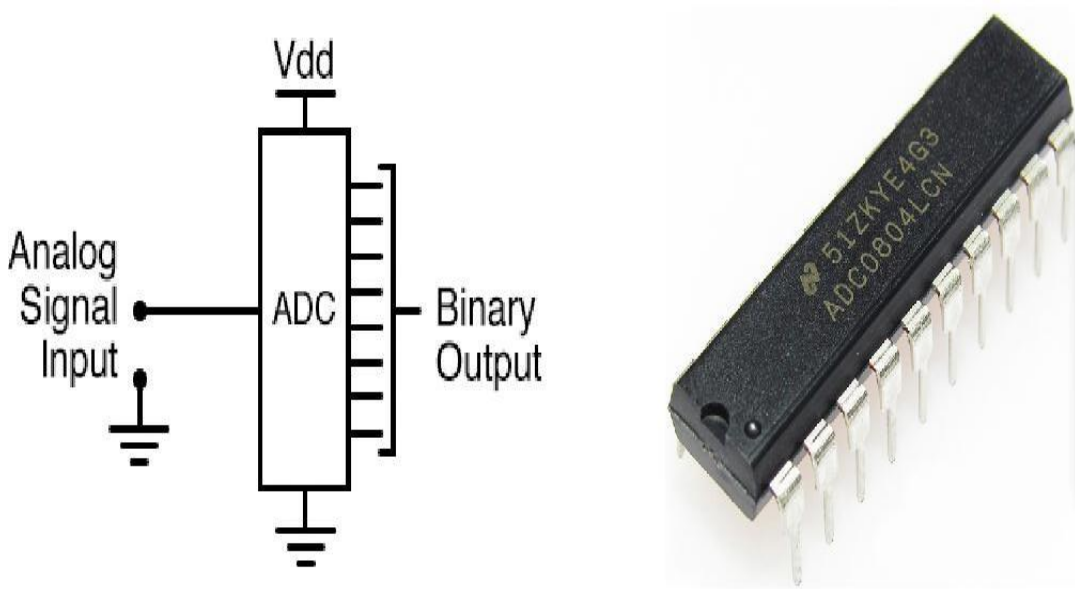
Membaca Potensiometer (Analog) dengan 8051

Kita hidup di dunia analog, dikelilingi oleh perangkat digital. Segala sesuatu yang kita lihat, rasakan, atau ukur bersifat analog seperti cahaya, suhu, kecepatan, tekanan, dll. Tetapi sebagian besar perangkat elektronik di sekitar kita mulai dari jam tangan digital sederhana hingga komputer super semuanya adalah perangkat digital. Jadi, kita sering perlu mengukur dan memproses sinyal yang bervariasi; ini

disebut sinyal analog. Jadi, jelas bahwa kita memerlukan sesuatu yang dapat mengubah parameter analog ini menjadi nilai digital agar mikrokontroler atau mikroprosesor dapat memahaminya. Sesuatu ini disebut ADC atau analog to digital converter. Sirkuit ADC ini dapat ditemukan sebagai ADC individu

IC sendiri atau tertanam ke dalam mikrokontroler. Karena mikrokontroler 8051 tidak memiliki adc internal, kami akan menggunakan IC adc eksternal untuk melakukan hal yang sama. Pada artikel ini kita akan mempelajari cara membaca sinyal analog menggunakan mikrokontroler 8051.

Pembahasan Perangkat Keras : IC ADC AD0804



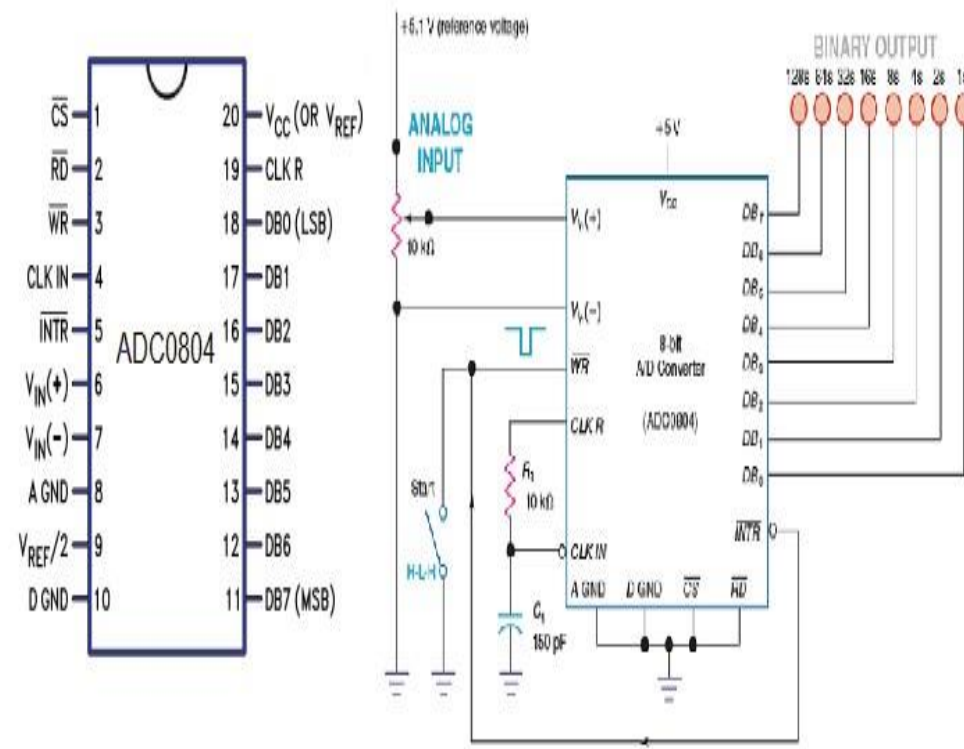
Gambar 9.14. diagram ADC

Gambar: Konversi Analog ke Digital dan Chip ADC AD0804 FITUR ADC 0804 :

- resolusi 8 bit
- Input tegangan analog diferensial
- Rentang tegangan input 0-5V (Tidak ada penyesuaian nol).
- Generator jam bawaan
- Tegangan pada $V_{ref}/2$ (pin9) dapat diatur secara eksternal
- mengkonversi rentang tegangan input yang lebih kecil ke resolusi 8 bit penuh.

- Modul ini tersedia dalam dua puluh pinouts kemasan PDIP (dual inline packaging) dan SOIC (small outline integrated circuits).
- Modul ini dapat dengan mudah terhubung dengan mikrokontroler lain dan juga dapat bekerja sendiri di sirkuit apa pun.

DESKRIPSI PIN: AD0803/AD0804/MCPXX



Gambar 9.15. Diagram rangkaian *AD0804 pin out*

- PIN-1 – CS (pilih chip) Ini adalah pin aktif rendah dan digunakan untuk mengaktifkan ADC0804. Ini adalah fitur hemat energi. Dalam contoh kita, kita akan menghubungkan pin ini ke ground.
- PIN-2 – RD (Baca) Merupakan pin input dan aktif rendah. ADC menyimpan hasilnya dalam register internal setelah konversi data analog. Pin ini membantu mengeluarkan data dari ADC0804. Ketika CS=0, pulsa tinggi ke rendah diberikan ke pin RD, maka keluaran digital datang pada pin D0-D7.
- PIN-3 – WR (Write) Merupakan pin input dan aktif low yang digunakan untuk menginisiasi ADC untuk memulai proses konversi. Ketika CS=0, WR melakukan transisi rendah ke tinggi, maka ADC memulai proses konversi.
- PIN-4 – Jam Masuk (CLK IN) Hubungkan jam eksternal ke pin ini.

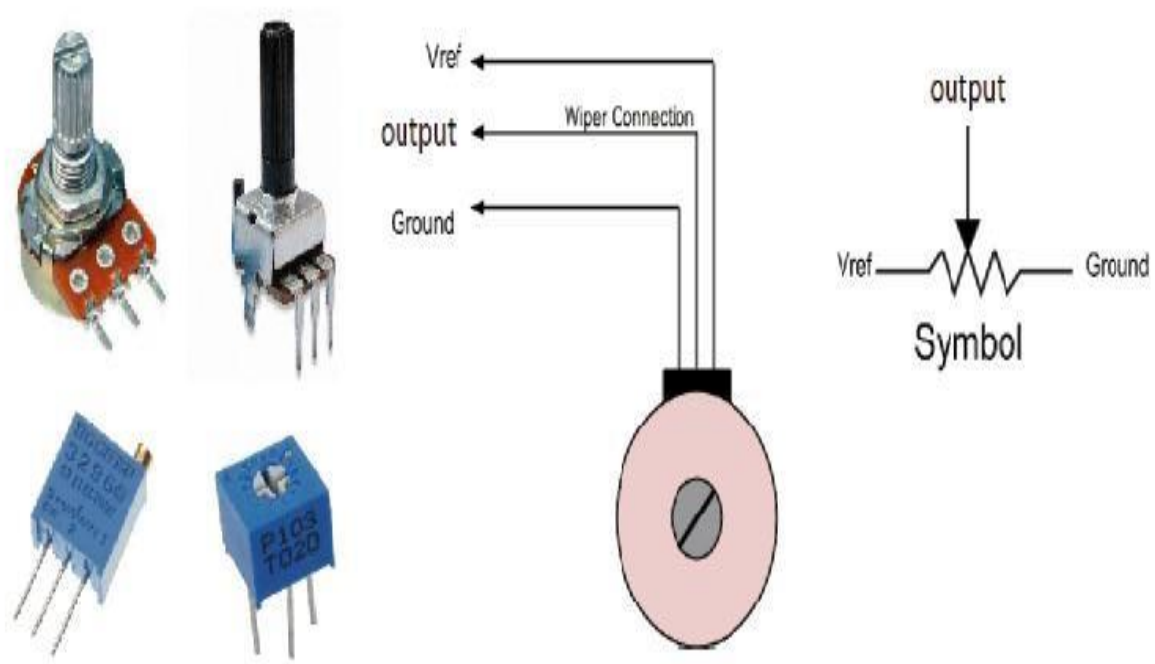
- PIN-5 – INTR (Interrupt) Pin ini secara otomatis menjadi rendah ketika konversi dilakukan oleh ADC0804 atau ketika input analog analog siap.
- PIN-6 – Vin + menghubungkan pin sensor analog input/tegangan input ke pin ini.
- PIN-7 – Vin - mengardekan pin ini.
- PIN-8 – AGND (Analog Ground) mengardekan pin ini.
- PIN-9 – Vref/2 mengatur tegangan referensi untuk input analog (Pin yang sangat penting dibahas di bawah)
- PIN-10 – DGND (Digital Ground) menghubungkan pin ini
- PIN-11 hingga PIN-18 – Data keluar*pin masukan.
- PIN-19 – CLK R (Jam R) Digunakan dengan pin clock IN saat sumber clock internal digunakan. • PIN-20 – Vcc +5 volt

PIN DESCRIPTION: AD0803/AD0804/MCPXX

- CS (pilih chip) rendah(0) untuk mengaktifkan adc0804.
- Pin WR (tuliskan) rendah (0).
- WR (tuliskan) pin tinggi (1) setelah beberapa penundaan penundaan kecil. Impuls rendah ke tinggi pada pin WR ini memulai konversi Anda.
- Sekarang periksa pin INTR(interrupt) jika high(1) konversi berjalan jika low(0) konversi selesai.
- RD =rendah(0) dan setelah beberapa waktu tinggi(1). Ini akan membawa nilai yang dikonversi ke 8 pin output data ADC 0804.

Variable Resistor :

Kami telah bekerja dengan resistor biasa sebelumnya, dan resistor variabel melakukan hal yang hampir sama. Satu-satunya perbedaan di sini adalah kita dapat mengubah nilai resistansi resistor variabel. Ada banyak cara berbeda untuk mengoperasikannya - mulai dari dial hingga slider dan segala macam di antaranya. Mereka digunakan sebagai sensor, pengatur, dan mekanisme pemicu. Bagi kami, kami akan menggunakan resistor variabel berbasis dial yang disebut potensiometer untuk bertindak sebagai tombol penyesuaian untuk LED.



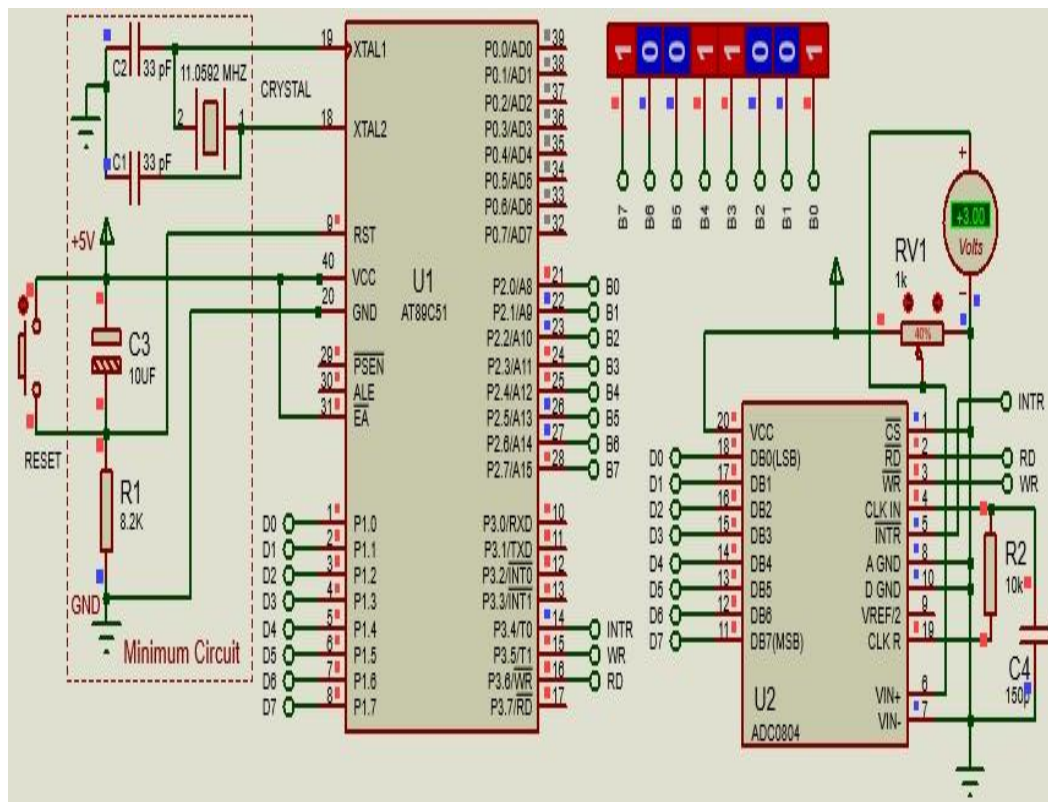
Gambar 9.16. Variabel resistor dan potensiometer

Variabel Resistor:

Potensiometer: Potensiometer pemangkas, juga dikenal sebagai trimpot, pada dasarnya adalah dua resistor variabel (R_1 dan R_2) yang dihubungkan secara seri. Resistansi total dari dua resistor variabel ($R_1 + R_2$) akan selalu sama dengan nilai trimpot, dalam kasus kami 10K. Namun, kita dapat memutar kenop pada trimpot untuk mengurangi resistansi satu resistor dan pada saat yang sama meningkatkan resistansi resistor lainnya. Jika kita memutar kenop ke kedua ujungnya, satu resistor akan menjadi 0R dan yang lainnya akan menjadi 10K.

Potensiometer memiliki kisaran resistansi. Mereka dapat disetel dari nol ohm ke resistansi maksimum apa pun yang khusus untuknya. Misalnya, potensiometer 10 k dapat disesuaikan dari 0 ohm hingga maksimum 10 k.

Circuit Diagram :



Gambar 9.17. ADC 0804 dengan AT89S51

Diskusi Sirkuit :

Port 1 digunakan sebagai port input dan port 0 digunakan sebagai port output. Potensiometer digunakan untuk mengatur tegangan. Jika tegangan input analog adalah 5V maka semua LED akan menyala menunjukkan 11111111 dalam biner yang setara dengan 255 dalam desimal. Jika tegangannya nol, tidak ada LED yang menyala.

Ketika tegangan penuh diberikan: (Output = 11111111)

Diskusi Pasca Jalankan: LED Chaser (Optimalisasi Kode)

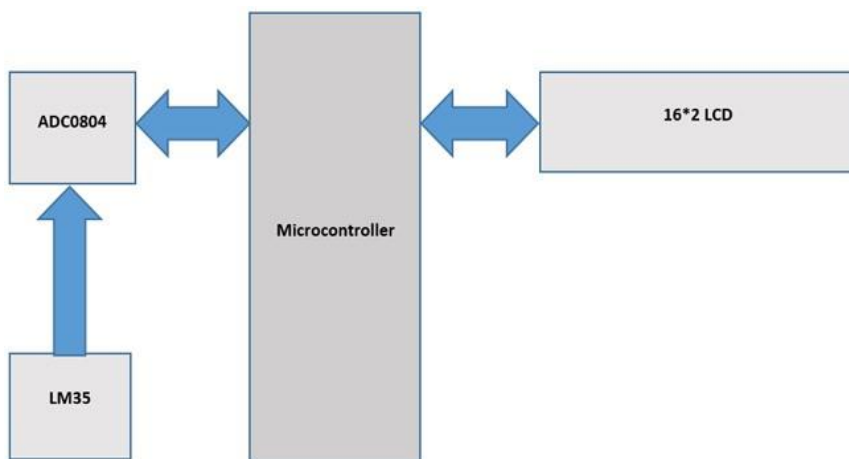
LM35 adalah analog, sensor suhu linier yang tegangan outputnya bervariasi secara linier dengan perubahan suhu. LM35 adalah sensor suhu linier tiga terminal dari semikonduktor Nasional. Dapat mengukur suhu dari -55 derajat celsius hingga $+150$ derajat celsius. Tegangan keluaran LM35 meningkat 10mV per derajat Celcius kenaikan suhu.

LM35 dapat dioperasikan dari suplai 5V dan arus siaga kurang dari 60uA. Pin keluar dari LM35 ditunjukkan pada gambar di bawah ini.



Gambar 9.18. Pin LM35 keluar

Jadi itu semua info yang Anda butuhkan tentang LM35 untuk proyek tampilan suhu khusus ini menggunakan mikrokontroler. Jadi mari kita beralih ke sensor suhu LM35 yang berinteraksi dengan 8051 kami



Gambar 9.19. Menghubungkan ADC dengan Mikroprosesor 8085

Pada bagian ini kita akan melihat bagaimana ADC (Analog to Digital Converter) bekerja dengan Mikroprosesor Intel 8085. Konversi Analog ke Digital adalah proses kuantisasi. Di sini sinyal analog diwakili oleh keadaan biner yang setara.

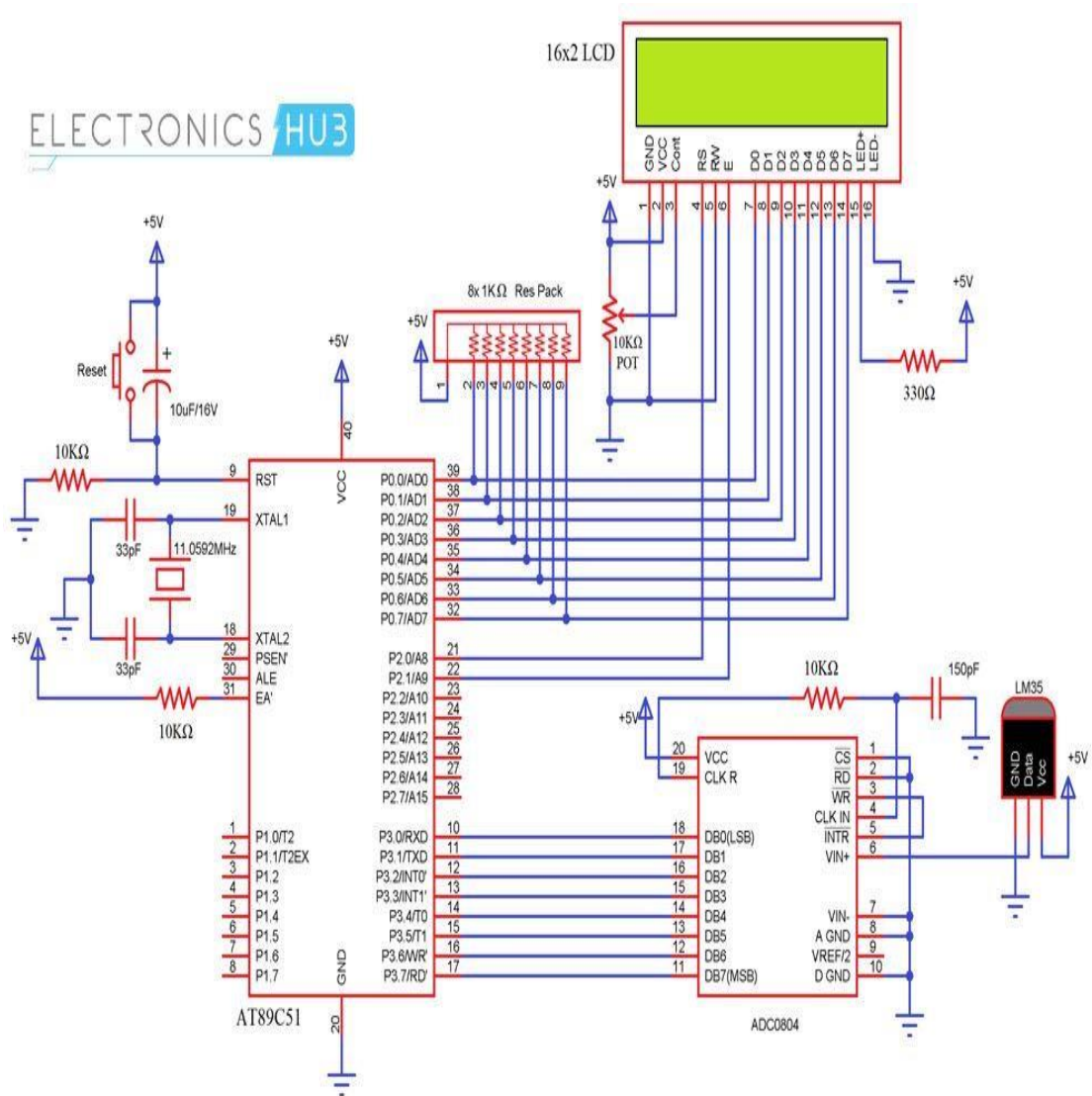
Konverter A/D dapat diklasifikasikan menjadi dua kelompok berdasarkan teknik konversinya.

Pada teknik pertama membandingkan sinyal analog yang diberikan dengan sinyal ekuivalen yang awalnya dihasilkan. Dalam teknik ini, ini mencakup aproksimasi berturut-turut, penghitung, dan konverter tipe flash. Dalam teknik lain menentukan perubahan sinyal analog menjadi waktu atau frekuensi. Proses ini mencakup konverter-integrator dan konverter tegangan-ke-frekuensi. Proses pertama lebih cepat tetapi kurang akurat, yang kedua lebih akurat. Karena proses pertama menggunakan tipe flash, jadi mahal dan sulit untuk mendesain untuk akurasi tinggi

Chip ADC 0808/0809

ADC 0808/0809 adalah konverter analog ke digital 8-bit. Ini memiliki 8 channel multiplexer untuk antarmuka dengan mikroprosesor.

Chip ini adalah ADC yang populer dan banyak digunakan. ADC 0808/0809 adalah perangkat CMOS monolitik. Perangkat ini menggunakan teknik aproksimasi berurutan untuk mengubah sinyal analog ke bentuk digital. Salah satu keuntungan utama dari chip ini adalah tidak memerlukan penyesuaian nol eksternal dan skala penuh, hanya pasokan +5V DC yang cukup.

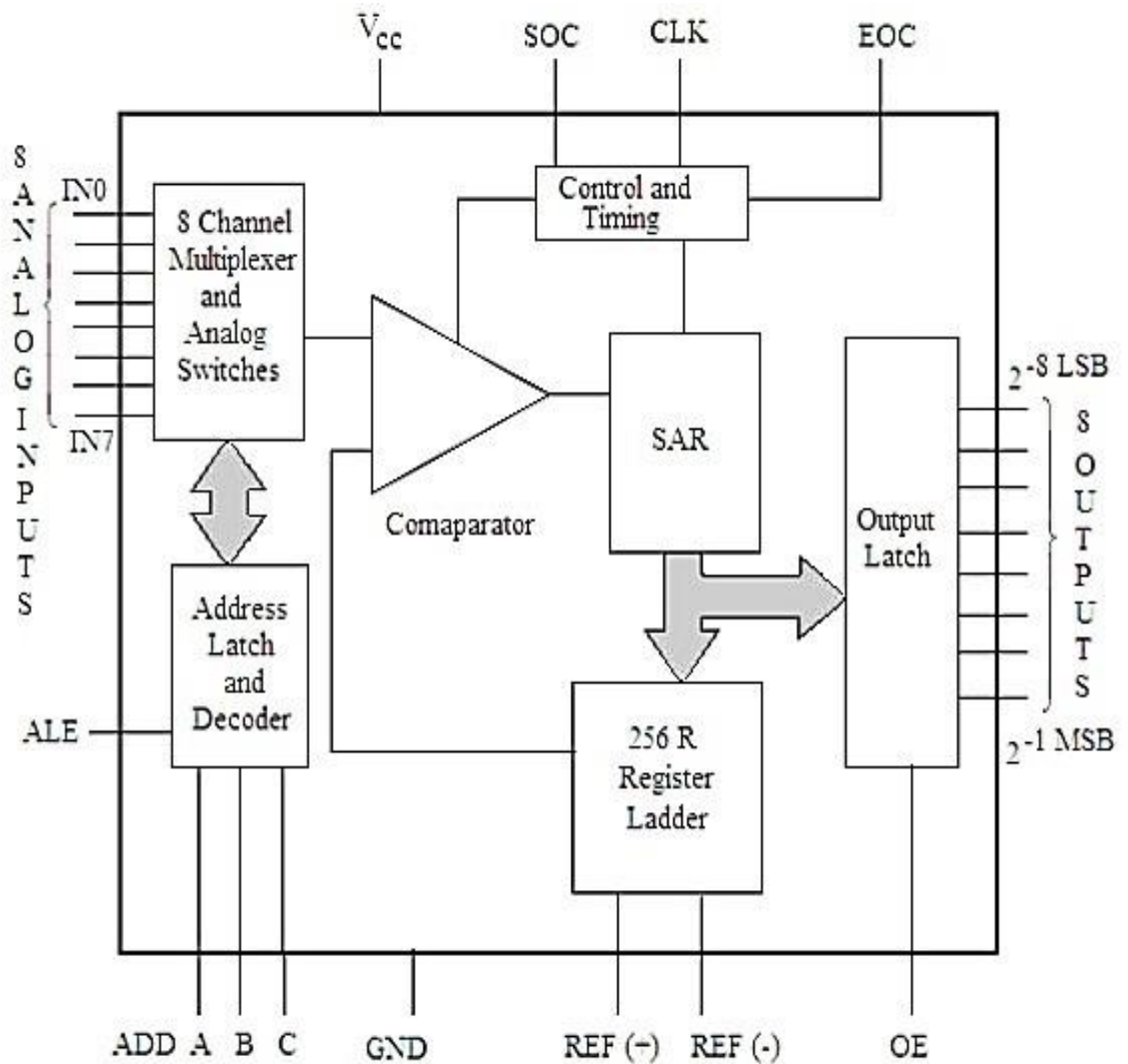


Gambar 9.20. Rangkaian mikrokontroler dengan ADC

Mari kita lihat beberapa fitur bagus dari ADC 0808/0809

- Kecepatan konversi jauh lebih tinggi
- Akurasi juga tinggi
- Memiliki ketergantungan suhu minimal
- Akurasi dan pengulangan jangka panjang yang sangat baik
- Konsumsi daya lebih sedikit

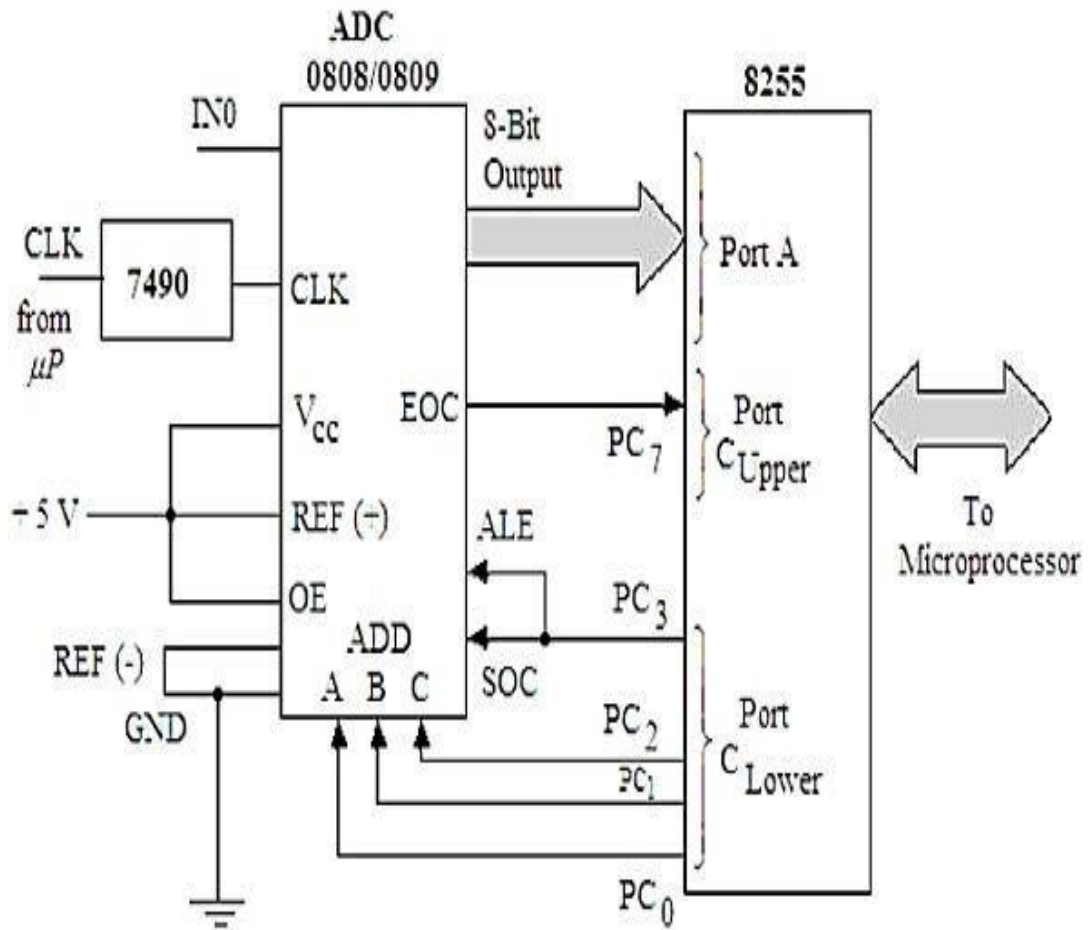
Diagram blok fungsional dari chip ini adalah seperti ini



Gambar 9.21. Diagram blok ADC 0808/0809

menghubungkan ADC dengan Mikroprosesor 8085

Untuk menghubungkan ADC dengan 8085, kita membutuhkan 8255 chip Antarmuka Periferal yang Dapat Diprogram dengannya. Mari kita lihat diagram rangkaian menghubungkan 8085, 8255 dan konverter ADC.



Gambar 9.22. Diagram interface ADC

Chip PortA 8255 digunakan sebagai port input. Pin PC7 dari Port Cupper terhubung ke Pin End of Conversion (EOC) dari konverter analog ke digital. Port ini juga digunakan sebagai port input. Port Clower digunakan sebagai port output. Jalur PC2-0 terhubung ke tiga pin alamat chip ini untuk memilih saluran input. Pin PC3 dihubungkan ke pin Start of Conversion (SOC) dan pin ALE ADC 0808/0809.

Sekarang mari kita lihat program untuk menghasilkan sinyal digital dari data analog. Kami menggunakan IN0 sebagai pin input, sehingga nilai pemilihan pin adalah 00H.

Arduino Programming

C++

Secara sederhana, sketch dalam Arduino dikelompokkan menjadi 3 blok (Gambar 3.2.), yakni :

Header, Setup

Loop.

Header

Pada bagian ini biasanya ditulis definisi-definisi penting yang akan digunakan selanjutnya dalam program, misalnya penggunaan library dan pendefinisian variable. Code dalam blok ini dijalankan hanya sekali pada waktu compile. Di bawah ini contoh code untuk mendeklarasikan variable led (integer) dan sekaligus di isi dengan angka 13 `int led = 13;`

Setup

Di sinilah awal program Arduino berjalan, yaitu di saat awal, atau ketika power on Arduino board. Biasanya di blok ini diisi penentuan apakah suatu pin digunakan sebagai input atau output, menggunakan perintah `pinMode`. Inisialisasi variable juga bisa dilakukan di blok ini // the setup routine runs once when you press reset: `void setup() { // initialize the digital pin as an output. pinMode(led, OUTPUT); }` `OUTPUT` adalah suatu makro yang sudah didefinisikan Arduino yang berarti = 1. Jadi perintah di atas sama dengan `pinMode(led, 1);` Suatu pin bisa difungsikan sebagai `OUTPUT` atau `INPUT`. Jika difungsikan sebagai output, dia siap mengirimkan arus listrik (maksimum 100 mA) kepada beban yang disambungkannya. Jika difungsikan sebagai `INPUT`, pin tersebut memiliki impedance yang tinggi dan siap menerima arus yang dikirimkan kepadanya.

Loop

Blok ini akan dieksekusi secara terus menerus. Apabila program sudah sampai akhir blok, maka akan dilanjutkan dengan mengulang eksekusi dari awal blok. Program akan berhenti apabila tombol power Arduino di matikan. Di sinilah fungsi utama program Arduino kita berada.

```
void loop() { digitalWrite(led, HIGH); // nyalakan LED
delay(1000); // tunggu 1000 milidetik
digitalWrite(led, LOW); // matikan LED delay(1000);
// tunggu 1000 milidetik }
```


Perintah `digitalWrite(pinNumber,nilai)` akan memerintahkan arduino untuk menyalakan atau mematikan tegangan di `pinNumber` tergantung nilainya. Jadi perintah di atas `digitalWrite(led,HIGH)` akan membuat pin nomor 13 (karena di header dideklarasikan `led = 13`) memiliki tegangan = 5V (HIGH). Hanya ada dua kemungkinan nilai `digitalWrite` yaitu HIGH atau LOW yang sebetulnya adalah nilai integer 1 atau 0

B. Program Sederhana Arduino Program sederhana adalah Led Blink, program ini akan mengakses pin 10 dan memerintahkan Arduino untuk mengulang blink led, - LED Flasher

```
int ledPin = 10; Sketch // Project 1 - LED Flasher
int ledPin = 10; void setup() { pinMode(ledPin, OUTPUT); }
void loop() { digitalWrite(ledPin, HIGH); delay(1000); digitalWrite(ledPin, LOW); delay(1000); loop() { digitalWrite(ledPin, HIGH); delay(1000); digitalWrite(ledPin, LOW); delay(1000); }
```

Pembahasan Sketch // Project 1 - LED Flasher Ini adalah komentar baris yang berguna untuk dokumentasi program, kompiler akan mengabaikan bagian ini. Baris komentar berguna bagi programmer agar bisa mengerti maksud program. `int ledPin = 10;` Inisialisasi variable, dalam hal ini inisialisasi variable bernama `ledPin` dengan type data integer dan nilai 10. `void setup() { pinMode(ledPin, OUTPUT); }` Setiap sketch Arduino wajib memiliki fungsi `setup()` dan `loop()`. Fungsi `setup()` dipanggil hanya sekali saat pertama kali program berjalan. Fungsi `setup()` biasanya tempat untuk men-setup hal-hal umum agar program siap dijalankan, seperti setup pin modes, setting serial baud rates, dan lainnya. Pada sketch Led Blink, fungsi `setup` hanya memiliki 1 baris perintah yaitu `pinMode(ledPin, OUTPUT);` `pinMode` fungsi yang berguna untuk memberitahu arduino bahwa pin pada board akan digunakan sebagai input atau output. Dalam baris program di atas, memberitahu arduino untuk menset pin 10 (nilai `ledPin` adalah 10) sebagai Output. `void loop() { digitalWrite(ledPin, HIGH); delay(1000); digitalWrite(ledPin, LOW); delay(1000); }` Fungsi `loop()` function adalah program utama yang dipanggil secara continue selama Arduino menyala. Setiap perintah dalam fungsi `loop()` akan dipanggil satu persatu sampai perintah terakhir dalam blok `loop` dicapai, lalu Arduino akan kembali ke awal perintah di blok fungsi `loop()`, sampai Arduino dimatikan atau tombol reset ditekan.

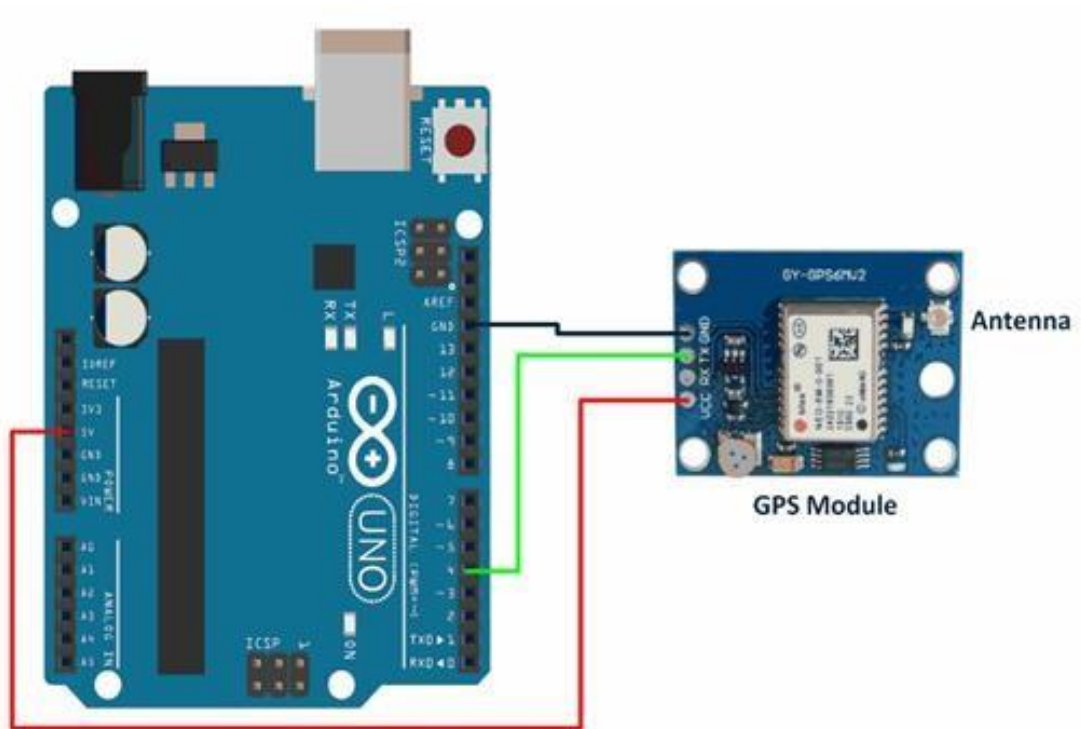
BAB X Modul Interfacing

Modul Interfacing GPS Dengan Arduino UNO



Gambar 9.23 Modul Penerima GPS NEO-6M

Global Positioning System (GPS) memanfaatkan sinyal yang dikirim oleh satelit di ruang angkasa dan stasiun bumi di Bumi untuk secara akurat menentukan posisinya di Bumi. Modul penerima GPS NEO-6M menggunakan komunikasi USART untuk berkomunikasi dengan mikrokontroler atau terminal PC. Ia menerima informasi seperti lintang, bujur, ketinggian, waktu UTC, dll. dari satelit dalam bentuk string NMEA. String ini perlu diurai untuk mengekstrak informasi yang ingin kita gunakan. Untuk informasi lebih lanjut tentang GPS dan cara menggunakannya, lihat topik Modul Penerima GPS di bagian sensor dan modul.



Gambar 9.24 Rangkaian Modul Penerima GPS NEO-6M Dengan Arduino UNO

Jika Anda melihat sebagian besar data dalam format *****, ambil GPS yang terhubung ke Arduino di ruang terbuka (balkon misalnya). GPS mungkin memerlukan beberapa waktu untuk mengunci satelit. Berikan sekitar 20-30 detik sehingga dapat mulai memberi Anda data yang benar. Biasanya diperlukan waktu tidak lebih dari 5 detik untuk mengunci satelit jika Anda berada di ruang terbuka, tetapi terkadang diperlukan waktu lebih lama (misalnya jika 3 satelit atau lebih tidak terlihat oleh penerima GPS).

Skets Untuk Menampilkan Parameter GPS Pada Serial Monitor

```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
/* Create object named bt of the class SoftwareSerial */
SoftwareSerial GPS_SoftSerial(4, 3); /* (Rx, Tx) */
/* Create an object named gps of the class TinyGPSPlus */
TinyGPSPlus gps;
volatile float minutes, seconds; volatile
int degree, secs, mins;
void setup() {
  Serial.begin(9600); /* Define baud rate for serial communication */
  GPS_SoftSerial.begin(9600); /* Define baud rate for software serial
communication */
} void loop()
{
```

```

        smartDelay(1000);      /* Generate precise delay of 1ms
*/      unsigned long start;      double lat_val, lng_val,
alt_m_val;      uint8_t hr_val, min_val, sec_val;      bool
loc_valid, alt_valid, time_valid;
        lat_val = gps.location.lat();      /* Get latitude data */
        loc_valid = gps.location.isValid(); /* Check if valid location data is available */
        lng_val = gps.location.lng();      /* Get longitude data */
alt_m_val = gps.altitude.meters(); /* Get altitude data in meters */
alt_valid = gps.altitude.isValid(); /* Check if valid altitude data is
available */      hr_val = gps.time.hour(); /* Get hour */      min_val =
gps.time.minute(); /* Get minutes */      sec_val = gps.time.second();
/* Get seconds */
        time_valid = gps.time.isValid(); /* Check if valid time data is available */
if (!loc_valid)
    {
        Serial.print("Latitude : ");
        Serial.println("*****");
        Serial.print("Longitude : ");
        Serial.println("*****");
    }
else
    {
        DegMinSec(lat_val);
        Serial.print("Latitude in Decimal Degrees : ");
        Serial.println(lat_val, 6);
        Serial.print("Latitude in Degrees Minutes Seconds : ");
        Serial.print(degree);
        Serial.print("\t");
        Serial.print(mins);
        Serial.print("\t");
        Serial.println(secs);
        DegMinSec(lng_val); /* Convert the decimal degree value into degrees minutes
seconds form */
        Serial.print("Longitude in Decimal Degrees : ");
        Serial.println(lng_val, 6);
        Serial.print("Longitude in Degrees Minutes Seconds : ");
        Serial.print(degree);
        Serial.print("\t");
        Serial.print(mins);
        Serial.print("\t");
        Serial.println(secs);
    }    if
(!alt_valid)    {
        Serial.print("Altitude : ");
        Serial.println("*****");
    }
else

```

```

    {
        Serial.print("Altitude : ");
        Serial.println(alt_m_val, 6);
    }
    if (!time_valid)
    {
        Serial.print("Time : ");
        Serial.println("*****");
    }
else
    {
        char time_string[32];
        sprintf(time_string, "Time : %02d/%02d/%02d \n", hr_val, min_val, sec_val);
        Serial.print(time_string);
    }
}
static void smartDelay(unsigned long ms)
{
    unsigned long start = millis();
    do
    {
        while (GPS_SoftSerial.available())/* Encode data read from GPS while data is available
on serial port */
            gps.encode(GPS_SoftSerial.read());
        /* Encode basically is used to parse the string received by the GPS and to store it in
a buffer so that information can be extracted from it */
    } while (millis() - start < ms);
}

void DegMinSec( double tot_val)                /* Convert data in decimal degrees into
degrees minutes seconds form */
{
    degree = (int)tot_val;
    minutes = tot_val - degree;
    seconds = 60 * minutes;
    minutes = (int)seconds;  mins
= (int)minutes;  seconds =
seconds - minutes;  seconds =
60 * seconds;  secs =
(int)seconds;
}

```

Antarmuka Modul Bluetooth HC-05 dengan Arduino UNO

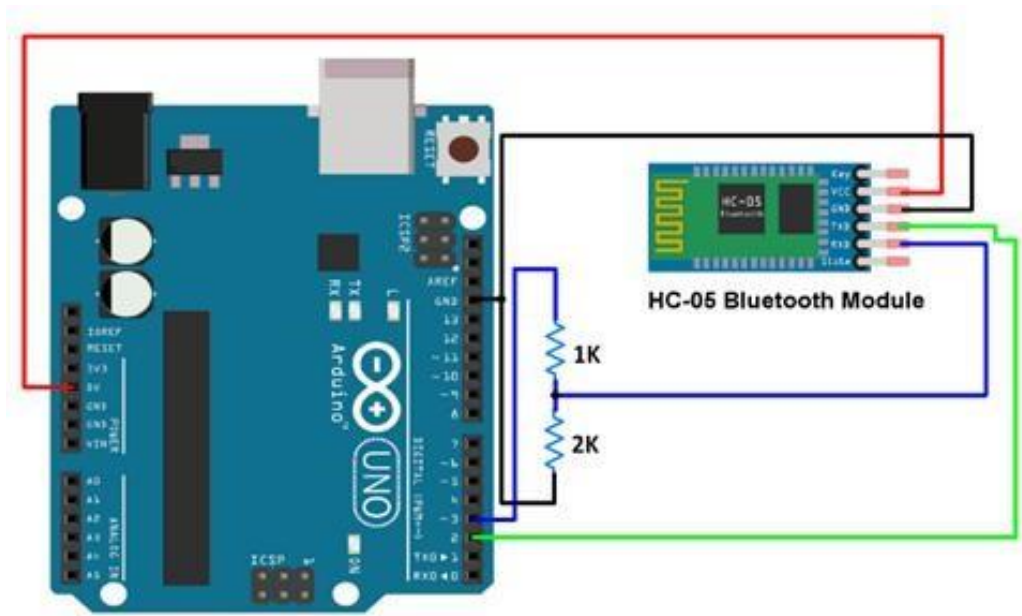


Gambar 9.26 Modul Bluetooth HC-05

HC-05 adalah perangkat Bluetooth yang digunakan untuk komunikasi nirkabel dengan perangkat berkemampuan Bluetooth (seperti smartphone). Ini berkomunikasi dengan mikrokontroler menggunakan komunikasi serial (USART). Pengaturan default modul Bluetooth HC-05 dapat diubah menggunakan perintah AT tertentu.

Karena modul Bluetooth HC-05 memiliki level 3,3 V untuk RX/TX dan mikrokontroler dapat mendeteksi level 3,3 V, sehingga modul HC-05 tidak perlu menggeser level tegangan TX. Tetapi kita perlu menggeser level tegangan pancar dari mikrokontroler ke RX modul HC-05.

Untuk informasi lebih lanjut tentang modul Bluetooth HC-05 dan cara menggunakannya, lihat topik Modul Bluetooth HC-05 di bagian sensor dan modul.



Gambar 9.27 Interfacing Module HC-05 Bluetooth dengan Arduino UNO

Catatan : Nama Bluetooth default perangkat adalah "HC-05" dan PIN (kata sandi) default untuk koneksi adalah "0000" atau "1234".

Sketch untuk Menampilkan Data yang Diterima Melalui Bluetooth Pada Serial Monitor

```

#include<SoftwareSerial.h>

/* Create object named bt of the class SoftwareSerial */
SoftwareSerial bt(2,3); /* (Rx,Tx) */ void setup() { bt.begin(9600); /*
Define baud rate for software serial communication */
Serial.begin(9600); /* Define baud rate for serial communication */
} void loop()
{
  if (bt.available()) /* If data is available on serial port */
  {
    Serial.write(bt.read()); /* Print character received on to the serial monitor */
  }
}

```

Interfacing Analog Joystick dengan Arduino UNO

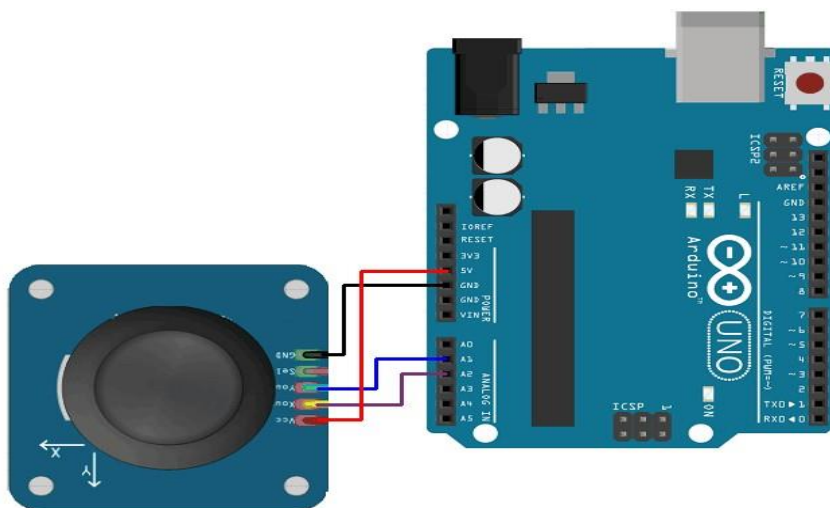


Aplikasi seperti video game yang memerlukan perubahan posisi kursor dalam bidang 2-D menggunakan joystick analog sebagai perangkat input.

Joystick analog menghasilkan dua tegangan; satu sesuai dengan posisi terhadap sumbu X dan yang lain sesuai dengan posisi terhadap sumbu Y. Tegangan yang dihasilkan tergantung pada posisi joystick.

Untuk informasi lebih lanjut tentang Analog Joystick dan cara menggunakannya, lihat topik Analog Joystick di bagian sensor dan modul.

Untuk menghubungkan Joystick Analog dengan Arduino Uno, kita perlu menggunakan ADC pada mikrokontroler Arduino UNO



Gambar 9.30 Interfacing Modul Analog Joystick dengan Arduino UNO
Menampilkan tegangan joystick analog dalam arah X dan Y pada serial monitor
Arduino.

Sketsa Sketch Untuk Menemukan Tegangan Joystick dalam arah X dan Y

```
const int joystick_x_pin = A2;
const int joystick_y_pin = A1;
```



```
void setup() {
  Serial.begin(9600); /* Define baud rate for serial communication */
} void loop() { int x_adc_val, y_adc_val; float x_volt, y_volt; x_adc_val
= analogRead(joystick_x_pin); y_adc_val = analogRead(joystick_y_pin);
x_volt = ( ( x_adc_val * 5.0 ) / 1023 ); /*Convert digital value to voltage */
y_volt = ( ( y_adc_val * 5.0 ) / 1023 ); /*Convert digital value to voltage */
  Serial.print("X_Voltage = ");
  Serial.print(x_volt);
  Serial.print("\t");
  Serial.print("Y_Voltage = ");
  Serial.println(y_volt); delay(100);
}
```

LM35 Interfacing with Arduino UNO



Gambar 9.31 Module sensor LM 35

LM35 adalah sensor suhu yang dapat mengukur suhu pada kisaran $-55\text{ }^{\circ}\text{C}$ hingga $150\text{ }^{\circ}\text{C}$.

Ini adalah perangkat 3-terminal yang memberikan tegangan analog sebanding dengan suhu. Semakin tinggi suhu, semakin tinggi tegangan output.

Output tegangan analog dapat diubah ke bentuk digital menggunakan ADC sehingga mikrokontroler dapat memprosesnya.

Untuk informasi lebih lanjut tentang LM35 dan cara menggunakannya, lihat topik Sensor Suhu LM35 di bagian sensor dan modul.

Mengukur suhu lingkungan menggunakan LM35 dan menampilkannya pada serial monitor Arduino. Di sini, output LM35 diberikan ke pin analog A1 Arduino UNO.

Tegangan analog ini diubah menjadi bentuk digitalnya dan diproses untuk mendapatkan pembacaan suhu.

Sketch untuk pengukuran suhu

```
const int lm35_pin = A1;    /* LM35 O/P pin
*/ void setup() { Serial.begin(9600);
} void loop() { int
temp_adc_val;
float temp_val;
temp_adc_val =
analogRead(lm35_
pin); /* Read
```

```
Temperature */
temp_val =
(temp_adc_val *
4.88); /* Convert
adc value to
equivalent voltage
*/
temp_val = (temp_val/10); /* LM35 gives output of 10mv/°C */
Serial.print("Temperature = ");
Serial.print(temp_val);
Serial.print("      Degree      Celsius\n");
delay(1000);
}
```

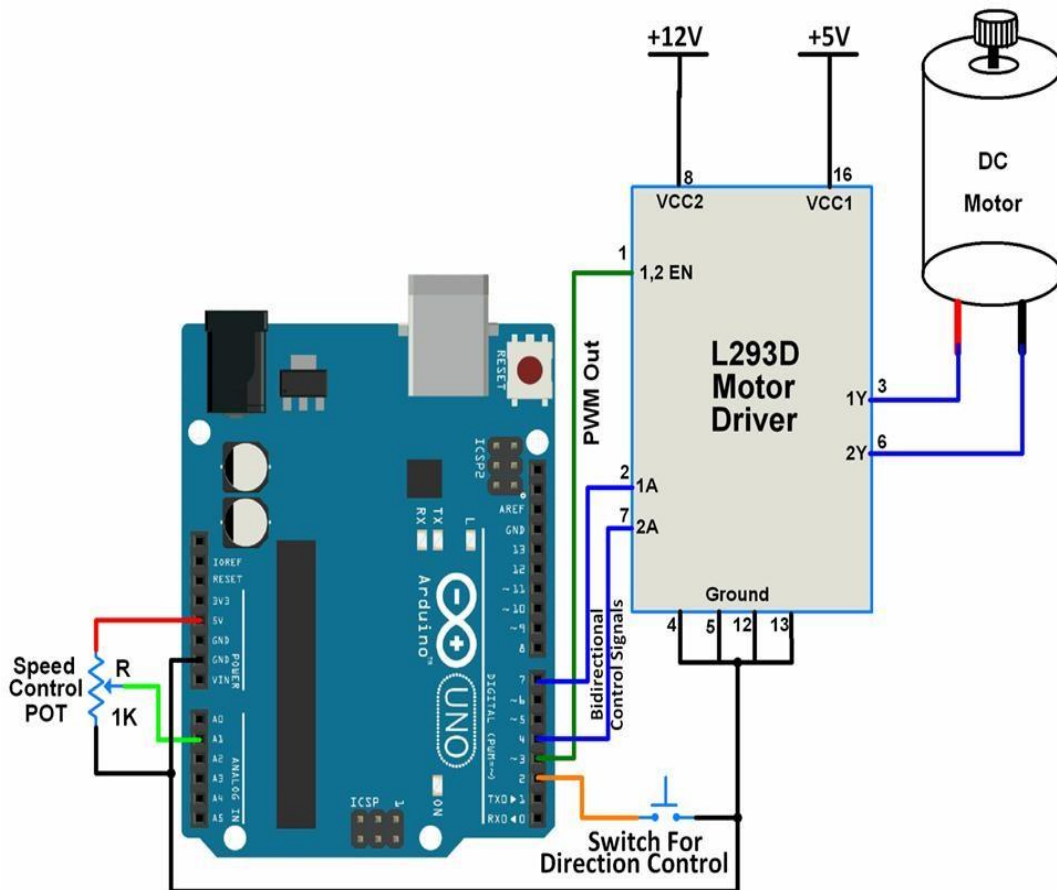
Antarmuka Motor DC dengan Arduino UNO



Gambar 9.32 Motor DC

Motor DC mengubah energi listrik berupa Arus searah menjadi energi mekanik berupa gerak rotasi poros motor. Kecepatan motor DC dapat dikontrol dengan menerapkan tegangan DC yang bervariasi; sedangkan arah putaran motor dapat diubah dengan membalik arah arus yang melaluinya.

Untuk menerapkan tegangan yang bervariasi, kita dapat menggunakan teknik PWM. Untuk membalikkan arus, kita dapat menggunakan rangkaian H-Bridge atau IC driver motor yang menggunakan teknik H-Bridge. Untuk informasi lebih lanjut tentang motor DC dan cara menggunakannya, konfigurasi rangkaian H-Bridge, dan teknik PWM, lihat topik Motor DC di bagian sensor dan modul.



Gambar 9.33 Rangkaian Motor DC dengan Arduino UNO

Disini kita akan mengatur kecepatan dan arah putaran motor DC menggunakan Arduino Uno. Di sini, potensiometer digunakan sebagai alat untuk mengontrol kecepatan dan input dari sakelar taktis digunakan untuk mengubah arah motor.

IC driver motor L293D digunakan untuk mengontrol arah motor. Gelombang PWM yang dihasilkan pada Arduino UNO digunakan untuk memberikan tegangan variabel ke motor melalui L293D. Di Arduino, fungsi analogWrite digunakan untuk membangkitkan gelombang PWM

Sketch program unjuk Arah dan Kontrol Kecepatan Motor DC

```
const int POT_input = A1; /* assign ADC Channel */
bool d1 = HIGH; bool d2 = LOW;
```

```
void setup() { pinMode(4, OUTPUT); /* Motor control pin 1 */ pinMode(7,
OUTPUT); /* Motor control pin 2 */ pinMode(3, OUTPUT); /* PWM pin for
```

```

Speed Control */ pinMode(2, INPUT_PULLUP); /* Interrupt pin for direction
control */ attachInterrupt(digitalPinToInterrupt(2), motor, FALLING); /*
Interrupt on falling edge on pin 2 */
}

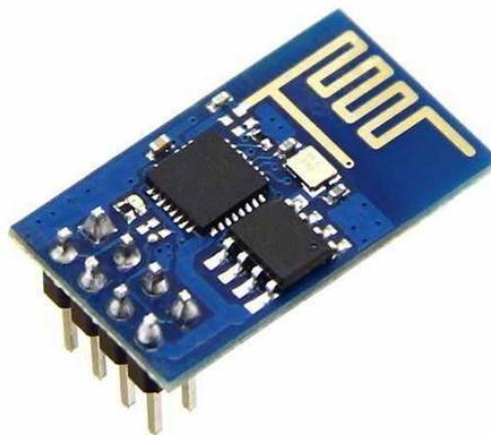
void loop() {
int pwm_adc;
pwm_adc = analogRead(POT_input); /* Input from Potentiometer for speed
control */ digitalWrite(4,d1); digitalWrite(7,d2);
analogWrite(3, pwm_adc / 4);
}

void motor(){
d1 = !d1; d2
= !d2;
delay_ms(200);
}

```

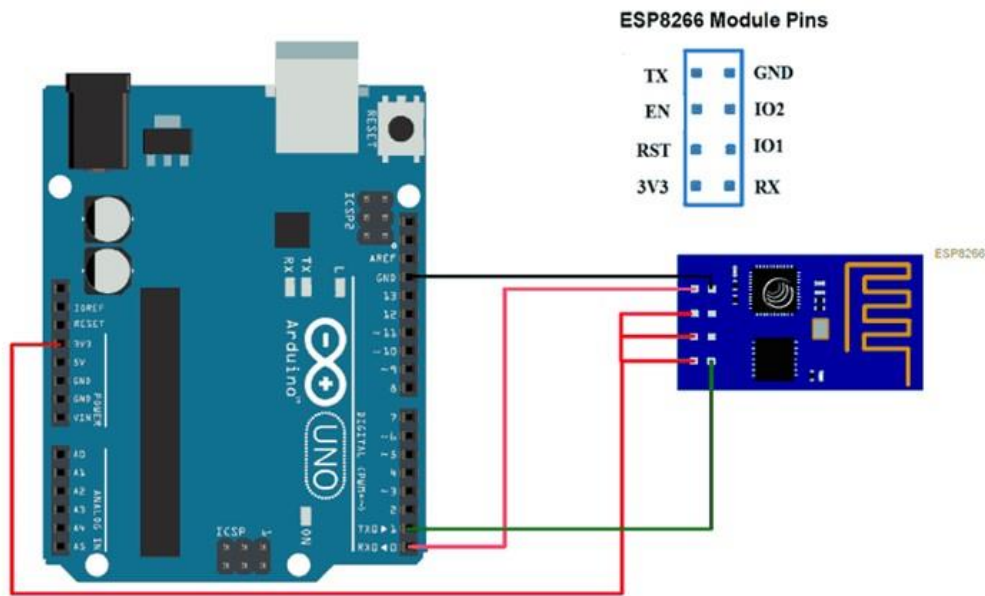
Antarmuka Modul WiFi ESP8266 dengan Arduino UNO

Modul wifi ESP8266 adalah transceiver nirkabel mandiri berbiaya rendah yang dapat digunakan untuk pengembangan IoT titik akhir. Modul wifi ESP8266 memungkinkan konektivitas internet ke aplikasi yang disematkan. Ini menggunakan protokol komunikasi TCP/UDP untuk terhubung dengan server/klien



Gambar 9.34 Modul Wi-Fi ESP8266

Untuk berkomunikasi dengan modul wifi ESP8266, mikrokontroler perlu menggunakan set perintah AT. Mikrokontroler berkomunikasi dengan modul wifi ESP8266-01 menggunakan UART yang memiliki Baud rate yang ditentukan (Default 115200).



Gambar 9.35. Interfacing diagram Wi-Fi ESP8266

Klien TCP menggunakan Modul WiFi ESP8266

Mari program Arduino UNO untuk mengkonfigurasi modul wifi ESP8266 sebagai TCP Client dan Menerima/Mengirim data dari/ke Server menggunakan WIFI.

Di sini, kami menggunakan server Thingspeak untuk tujuan demo TCP Client.

Thingspeak adalah platform IOT terbuka di mana siapa pun dapat memvisualisasikan dan menganalisis data langsung dari perangkat sensor mereka. Selain itu, kita dapat melakukan analisis data pada data yang diposting oleh perangkat jarak jauh dengan kode Matlab di Thingspeak. Untuk mempelajari lebih lanjut tentang tautan referensi Thingspeak

https://thingspeak.com/pages/learn_more Cukup daftar dan buat saluran. Kami memiliki saluran dan kunci tulis di bawah ini yang tersedia di Thingspeak untuk pengiriman dan penerimaan data.

ID saluran adalah = 119922

Kunci Tulis adalah = C7JFHZY54GLCJY38

Catatan: Jangan lupa untuk mencentang bidang Jadikan Publik di opsi pengaturan saluran di saluran thingspeak Anda. Itu membuat saluran tersedia untuk digunakan sebagai publik. Ini memungkinkan setiap pengguna untuk mengakses data saluran tanpa nama pengguna & kata sandi apa pun.

Untuk metode TCP RECEIVE gunakan di bawah langkah perintah AT yang ditunjukkan pada tangkapan layar Terminal Serial RealTerm.

```

RealTerm: Serial Capture Program 2.0.0.70
ATE0\r\n\r\n
OK\r\n
AT+CWMODE=3\r\n\r\n
OK\r\n
AT+CIPMUX=0\r\n\r\n
OK\r\n
AT+CIPMODE=0\r\n\r\n
OK\r\n
AT+CWJAP="EW-WIFI","mh163687"\r\n
WIFI DISCONNECT\r\n
WIFI CONNECTED\r\n
WIFI GOT IP\r\n
OK\r\n
AT+CIPSTART="TCP","api.thingspeak.com",80\r\n
CONNECT\r\n
OK\r\n
AT+CIPSEND=37\r\n\r\n
OK\r\n
> GET /channels/119922/feeds/last.txt\r\n\r\n
Recv 37 bytes\r\n
SEND OK\r\n
+IPD,66:<"created_at":"2017-07-07T13:36:33Z","entry_id":1117,"field1":"1">CLOSED\r\n

```

Gambar 9.36 Tangkapan layar

Tangkapan layar di bawah ini terdiri dari perintah AT (Hijau) dan Tanggapan (Kuning) Untuk metode TCP SEND gunakan di bawah langkah perintah AT yang ditunjukkan pada tangkapan layar Terminal Serial RealTerm.

```

RealTerm: Serial Capture Program 2.0.0.70
ATE0\r\n\r\n
OK\r\n
AT+CWMODE=3\r\n\r\n
OK\r\n
AT+CIPMUX=0\r\n\r\n
OK\r\n
AT+CIPMODE=0\r\n\r\n
OK\r\n
AT+CWJAP="EW-WIFI","mh163687"\r\n
WIFI DISCONNECT\r\n
WIFI CONNECTED\r\n
WIFI GOT IP\r\n
OK\r\n
AT+CIPSTART="TCP","api.thingspeak.com",80\r\n
CONNECT\r\n
OK\r\n
AT+CIPSEND=47\r\n\r\n
OK\r\n
> GET /update?api_key=C7JPHZY54GLCJY38&field1=1\r\n\r\n
Recv 47 bytes\r\n
SEND OK\r\n
+IPD,4:1117CLOSED\r\n

```


Untuk mengakses fungsionalitas yang diperlukan yaitu TCP menerima atau mengirim, buat perubahan dalam program seperti yang diberikan di bawah ini,

Untuk demo TCP Client MENERIMA

```
#define RECEIVE_DEMO    /* Define RECEIVE demo */
//#define SEND_DEMO     /* Define SEND demo */
Untuk demo KIRIM Klien TCP
```

```
//#define RECEIVE_DEMO /* Define RECEIVE demo */
#define SEND_DEMO     /* Define SEND demo */
Edit bidang di bawah ini dengan data masing-masing dalam file header ESP 8266
```

```
/* Define Required fields shown below */
#define DOMAIN        "api.thingspeak.com"
#define PORT          "80"
#define API_WRITE_KEY " Thingspeak Write Key "
#define CHANNEL_ID    " Thingspeak Channel ID "
#define SSID          " WiFi SSID "
#define PASSWORD      " WiFi Password "
```

Program

```
/*
 * ESP8266 wifi module Interfacing with Arduino Uno
 * http://www.electronicwings.com
 */
#include "ESP8266_AT.h"

/* Select Demo */
//#define RECEIVE_DEMO    /* Define RECEIVE demo */
#define SEND_DEMO       /* Define SEND demo */

/* Define Required fields shown below */
#define DOMAIN          "api.thingspeak.com"
#define PORT            "80"
#define API_WRITE_KEY   "Write your Write Key here"
#define CHANNEL_ID      "Write your Channel ID here"

#define SSID            "Write your WIFI SSID here"
#define PASSWORD        "Write your WIFI Password here"

char _buffer[150]; uint8_t
Connect_Status;
#ifdef SEND_DEMO
uint8_t Sample = 0;
```

```

#endif

void      setup()      {
Serial.begin(115200);

    while(!ESP8266_Begin());
    ESP8266_WIFIMode(BOTH_STATION_AND_ACCESPOINT);    /* 3 =
Both (AP and STA) */
    ESP8266_ConnectionMode(SINGLE);                    /* 0 = Single; 1 =
Multi */
    ESP8266_ApplicationMode(NORMAL);                  /* 0 = Normal
Mode; 1 = Transperant Mode */
    if(ESP8266_connected() == ESP8266_NOT_CONNECTED_TO_AP)/*Check
WIFI connection*/
    ESP8266_JoinAccessPoint(SSID, PASSWORD);          /*Connect to WIFI*/
    ESP8266_Start(0, DOMAIN, PORT);
}

void loop() {
    Connect_Status = ESP8266_connected();
    if(Connect_Status == ESP8266_NOT_CONNECTED_TO_AP) /*Again check
connection to WIFI*/
    ESP8266_JoinAccessPoint(SSID, PASSWORD);          /*Connect to WIFI*/
    if(Connect_Status == ESP8266_TRANSMISSION_DISCONNECTED)
    ESP8266_Start(0, DOMAIN, PORT);                    /*Connect to TCP
port*/

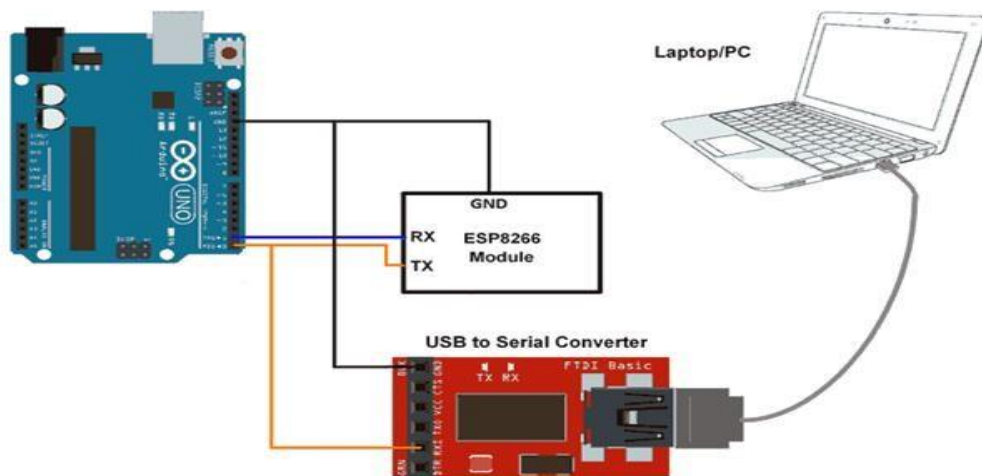
    #ifdef SEND_DEMO
    memset(_buffer, 0, 150);
    sprintf(_buffer, "GET /update?api_key=%s&field1=%d", API_WRITE_KEY,
Sample++); /*connect to thingspeak server to post data using your
API_WRITE_KEY*/
    ESP8266_Send(_buffer);
    delay(15000);                                       /* Thingspeak server delay */
    #endif

    #ifdef RECEIVE_DEMO
    memset(_buffer, 0, 150);
    sprintf(_buffer, "GET /channels/%s/feeds/last.txt", CHANNEL_ID); /*Connect
to thingspeak server to get data using your channel ID*/
    ESP8266_Send(_buffer);
    Read_Data(_buffer); delay(600);
    #endif
}

```

Tanggapan ESP8266

Di ujung klien, kita perlu memeriksa tanggapan ESP8266. Kita bisa mengeceknya di serial terminal PC/Laptop. Hubungkan pin pengirim (TX) modul ESP8266 ke pin penerima (RX) Arduino UNO dan pin penerima (RX) dari konverter USB ke serial seperti yang ditunjukkan pada gambar di bawah. sambungkan USB to serial converter ke PC/Laptop. Buka terminal serial pada PC/Laptop untuk melihat respon ESP8266 untuk perintah AT yang dikirim dari Arduino UNO.



Gambar 9.37 Rangkaian Arduino dengan Laptop

Sekarang untuk perintah TCP SEND (dikirim dari modul Arduino UNO), kita dapat melihat respon dari ESP8266 di bawah ini pada terminal serial untuk server thingspeak.

```

COM9 - Tera Term VT
File Edit Setup Control Window Help
STATUS:3
+CIPSTATUS:0,"TCP","52.1.229.129",80,41977,0
OK
OK
>
Recv 47 bytes
SEND OK
+IPD,4:1131CLOSED
STATUS:4
OK
+CIPMUX:0
OK
CONNECT
OK
OK
>
Recv 47 bytes
SEND OK
+IPD,4:1132CLOSED
STATUS:4
OK
+CIPMUX:0
OK
CONNECT
OK

```

Thingspeak responds with entry id for data send success

Gambar 3.38 Tangkapan layar tinkspeak saat kirim

Menanggapi dengan TCP KIRIM kami mendapatkan entri data no. seperti terlihat pada gambar di atas yaitu 1131, 1132, dan seterusnya.

Untuk perintah TCP RECEIVE (dikirim dari Modul Arduino UNO), kita dapat melihat respon ESP8266 di bawah ini pada terminal serial untuk server thingspeak.

```

COM9 - Tera Term VT
File Edit Setup Control Window Help
OK
CONNECT
OK
OK
>
Recv 37 bytes
SEND OK
+IPD,67:<"created_at":"2017-08-22T13:52:28Z","entry_id":1142,"field1":"11">CLOSED
STATUS:4
OK
+CIPMUX:0
OK
CONNECT
OK
OK
>
Recv 37 bytes
SEND OK
+IPD,67:<"created_at":"2017-08-22T13:52:28Z","entry_id":1142,"field1":"11">CLOSED
STATUS:4
OK
+CIPMUX:0

```

Thingspeak response for last updated data



Gambar 3.40 Tangkapan layar tingspeak saat terima

Sebagai tanggapan dengan TCP RECEIVE kami mendapatkan data entri terakhir untuk bidang 1 pada thingspeak seperti yang ditunjukkan pada gambar di atas.

Catatan: di sini kita mengambil data entri terakhir pada field1 dari server thingspeak sehingga kita mendapatkan data terakhir dari field1 dari server seperti yang ditunjukkan pada gambar di atas yaitu "field1":"11". Dalam program, kami menggunakan "GET /channels/119922/feeds/last.txt" untuk menerima data yang terakhir diperbarui saja.

Pembaruan di server thingspeak di TCP KIRIM

Untuk TCP SEND kita dapat melihat output di server end. Di sini kita menggunakan server thingspeak dan mengirimkan hitungan yang bertambah ke bidang 1 di server. Kami mendapatkan jumlah yang bertambah di field1 dari server thingspeak seperti yang ditunjukkan pada gambar di bawah ini.



Gambar 3.41 Tangkapan layar tingspeak

DAFTAR PUSTAKA

1. Frank Vahid and Tony Givargis, “Embedded System Design: A Unified Hardware/Software Approach”, 1999 • Ken Arnold, “Embedded Controller Hardware Design”, 2000 •
2. John J. Craig (1986), Introduction to Robotics, Pearson Education International
3. Charles M Bergen (2003), Anatomy of A Robot, McGraw Hill
4. <https://www.electronicwings.com/arduino/interfaces>
5. Arief Wisaksono¹ and Catur Aswar Ragil¹ “Design and Development of Parking Motor Parking Information System at Muhammadiyah University, Sidoarjo”,2020
6. Arief Wisaksono. Yanik Purwanti, Novia Ariyanti “Design of Monitoring and Control of Energy Use in Multi-storey Buildings based on IoT”,2020



UMSIDA PRESS
Universitas Muhammadiyah Sidoarjo
Jl. Mojopahit No. 666B
Sidoarjo, Jawa Timur

ISBN 978-623-464-050-2 (PDF)

