



Dasar Sistem Terbenam

Ali Idrus, S.Kom., M.Kom.

Muhammad Ridho Kurniawan Pratama, S.Kom., M.T.I.

Referensi Bahan Ajar

Embedded System and Robotics. (2017). *Universitas Negeri Makassar*. <https://eprints.unm.ac.id/4521/>



Pertemuan 1

Pengenalan Sistem Terbenam

Ali Idrus, S.Kom., M.Kom.

Muhammad Ridho Kurniawan Pratama, S.Kom., M.T.I.



Tujuan Pembelajaran

- 🎯 Menjelaskan apa itu Embedded System
- 📱 Memberikan contoh penggunaan Embedded System
- 📁 Menentukan klasifikasi Embedded System
- 🔧 Mengemukakan model perancangan Embedded System

Sistem Terbenam?

- 💻 Sistem komputer khusus untuk tugas tertentu, tertanam dalam sistem yang lebih besar
- ⚙️ Didukung mikrokontroler → berfungsi sebagai piranti kontrol
- 🧠 Dapat diprogram untuk kecerdasan buatan sederhana (AI)
- 📡 Dilengkapi input/output agar dapat berinteraksi dengan lingkungan



Apollo Guidance Computer – embedded system pertama di dunia

Klasifikasi Sistem Terbenam

 **Stand Alone** → dapat bekerja sendiri (contoh: MP3 player, kamera digital, video game)

 **Real-Time** → respon cepat sesuai batas waktu

- Hard Real-Time ( kritis, ex: kontrol rudal, airbag mobil)
- Soft Real-Time ( toleransi keterlambatan, ex: microwave, mesin cuci)

 **Networked** → terhubung dengan LAN/WAN/internet (contoh: sistem keamanan rumah)

Ciri Umum Sistem Terbenam

- 🎯 Dirancang untuk aplikasi khusus (single purpose)
- ⚡ Respon cepat sesuai kebutuhan real-time
- 🔧 Implementasi di perangkat keras maupun perangkat lunak
- 💰 Biaya relatif murah (produksi massal → efisiensi)
- 🔋 Konsumsi daya rendah agar tidak membebani sistem induk

Unsur Penting Sistem Terbenam

- ⌚ Real-time → respon sesuai tenggat waktu
- 🛡️ Aman bagi pengguna (safety) → ex: airbag mobil
- 🔒 Aman dari serangan (security)
- 🧩 Fault-tolerant → mampu menangani kegagalan sistem
- 🤖 Digunakan untuk fungsi khusus (contoh: robot, drone, pesawat)

Contoh Produk Sistem Terbenam

- 📱 Smartphone & kamera digital
- 🎮 Video game console & MP3 player
- 🏦 ATM, kartu pintar, terminal penjualan
- 🚗 Sistem biomedik, pengukur jarak mobil, airbag
- 📡 Router, switch, CCTV pintar
- 🏡 Sistem keamanan rumah (sensor, alarm, kamera)

Arsitektur

- **Perangkat keras** → unit kendali + datapath
- **Sistem operasi** (opsional, ex: RTOS, kernel sederhana)
- **Aplikasi** → fungsi tambahan di atas perangkat keras/OS

Teknologi Prosesor

- **General Purpose Processor** → fleksibel, contoh: PC, laptop
-  **Application-Specific Processor** → instruksi khusus untuk domain tertentu
-  **Special Purpose Processor** → untuk fungsi tunggal, contoh: kamera digital

Platform Pengujian

-  Mikroprosesor → kendali real-time (contoh: robot, motor)
-  Mikrokontroler → komponen sederhana, hemat biaya
-  ASSP (Application-Specific Standard Product) → fungsi tertentu di PCB
-  FPGA → mudah dikonfigurasi ulang, kecepatan tinggi, paralel

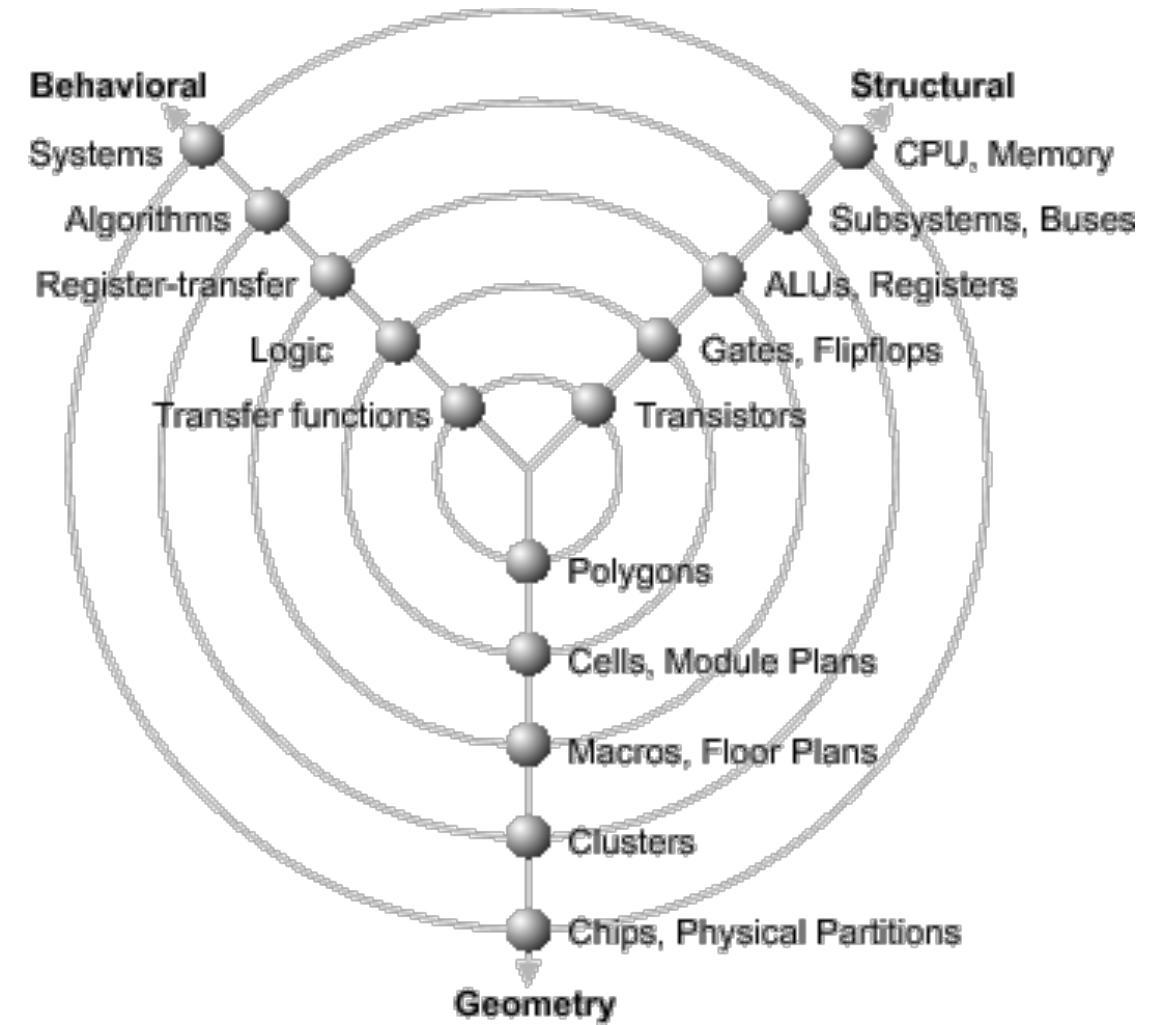
Tren Teknologi Sistem Terbenam

-  Perangkat cerdas (AI, fuzzy logic, machine learning)
-  Prosesor multi-core → kecepatan tinggi, throughput besar
-  Konvergensi perangkat (smartphone, tablet, IoT)
-  Teknologi nirkabel (WiFi, LTE, RFID, Bluetooth)
-  Open source OS (Embedded Linux, Symbian)
-  Keamanan data & protokol komunikasi

Model Pengembangan

-  **Y-Chart** (Gajski): perilaku, struktur, fisik
-  **Big-Bang**: coba-coba, pengembangan modular
-  **Waterfall**: tahapan berurutan (definisi → desain → implementasi → uji)
-  **Rapid Prototyping**: cepat, iteratif, fokus perbaikan kebutuhan pengguna
-  **Parallel Model**: HW & SW dirancang bersamaan untuk percepatan

Gajski-Kuhn Chart



Level & Framework Perancangan

- 💡 Level perancangan: transistor → gerbang → register transfer → sistem
- 🔧 Framework tools: Ptolemy, Cosyma, Vulcan, Stellar
- 🎯 Tujuan: optimasi biaya, kecepatan, reliabilitas

Ringkasan

- 💡 Embedded System = sistem komputer khusus tertanam dalam sistem lain
- ⚙️ Didukung prosesor, mikrokontroler, FPGA, dsb.
- ⌚ Real-time, aman, hemat biaya & energi
- 🤖 Banyak produk sehari-hari (smartphone, ATM, CCTV, kendaraan)
- 📈 Tren: perangkat cerdas, multi-core, nirkabel, open source, keamanan
- 🛠️ Model perancangan: Y-Chart, Waterfall, Rapid Prototyping, Parallel

Diskusi?



Pertemuan 2

Sistem Akuisisi Data

Ali Idrus, S.Kom., M.Kom.

Muhammad Ridho Kurniawan Pratama, S.Kom., M.T.I.

Referensi Bahan Ajar

Jaya, Hendra et al. (2017). Embedded System and Robotics.

Universitas Negeri Makassar. <https://eprints.unm.ac.id/4521/>

Di Paolo Emilio, M. (2013). Data Acquisition Systems. Springer New York. <https://doi.org/10.1007/978-1-4614-4214-1>

Tujuan Pembelajaran

- 🎯 Memahami prinsip dasar dari sistem akuisisi data pada sistem terbenam
- 🧩 Mengetahui berbagai komponen yang terlibat didalam sistem akuisisi data

Sistem Akuisisi Data

Proses utama Data Acquisition System: **melakukan *sampling* sinyal analog** dari kondisi fisik (tegangan ataupun arus listrik) dan **mengoversinya ke bentuk digital** yang dapat diolah oleh komputer

Elemen:

-  Sensor & transduser
-  Perkabelan
-  Rangkaian pengondisi sinyal
-  DAQ hardware (contoh: ADC, DAC)
-  DAQ software
-  Komputer

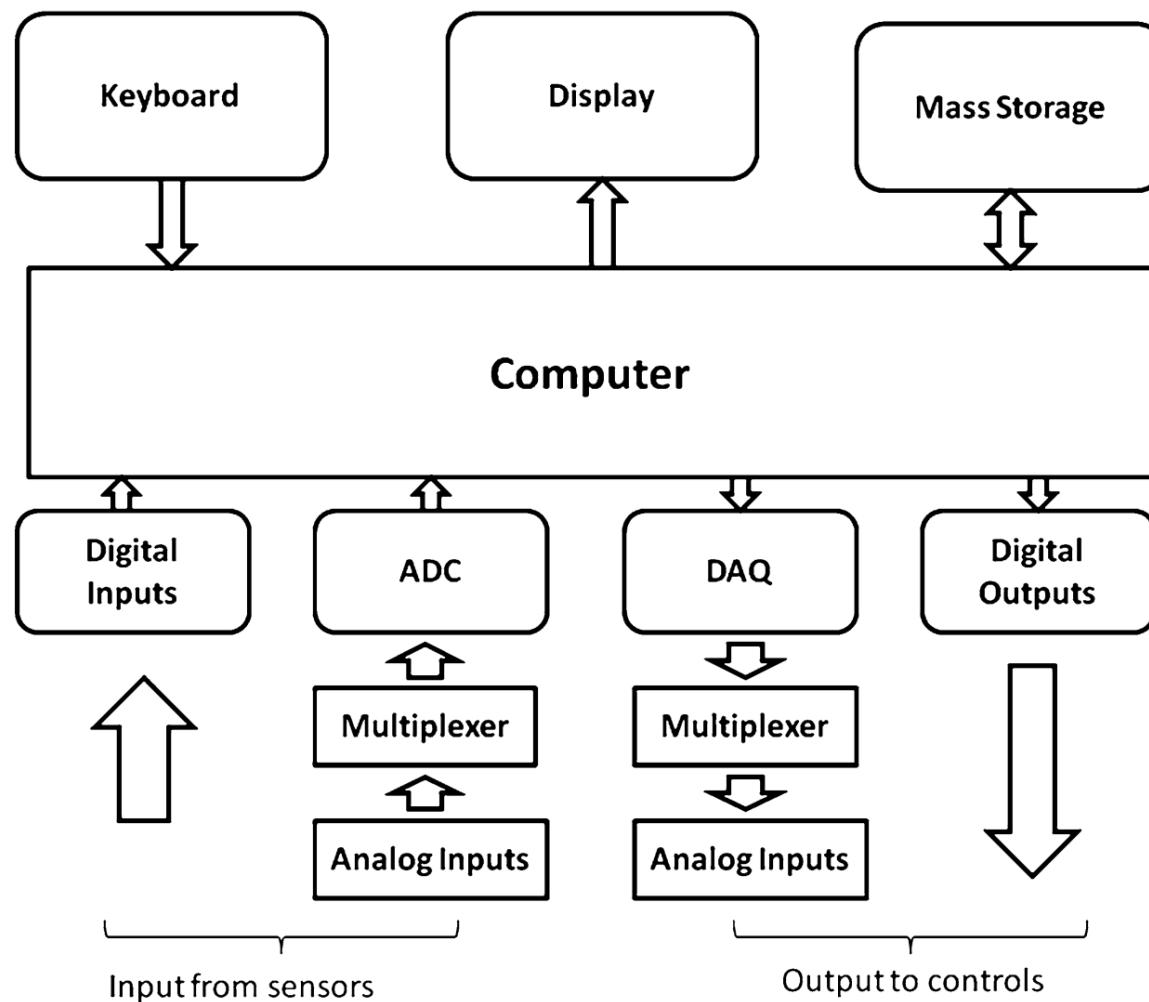
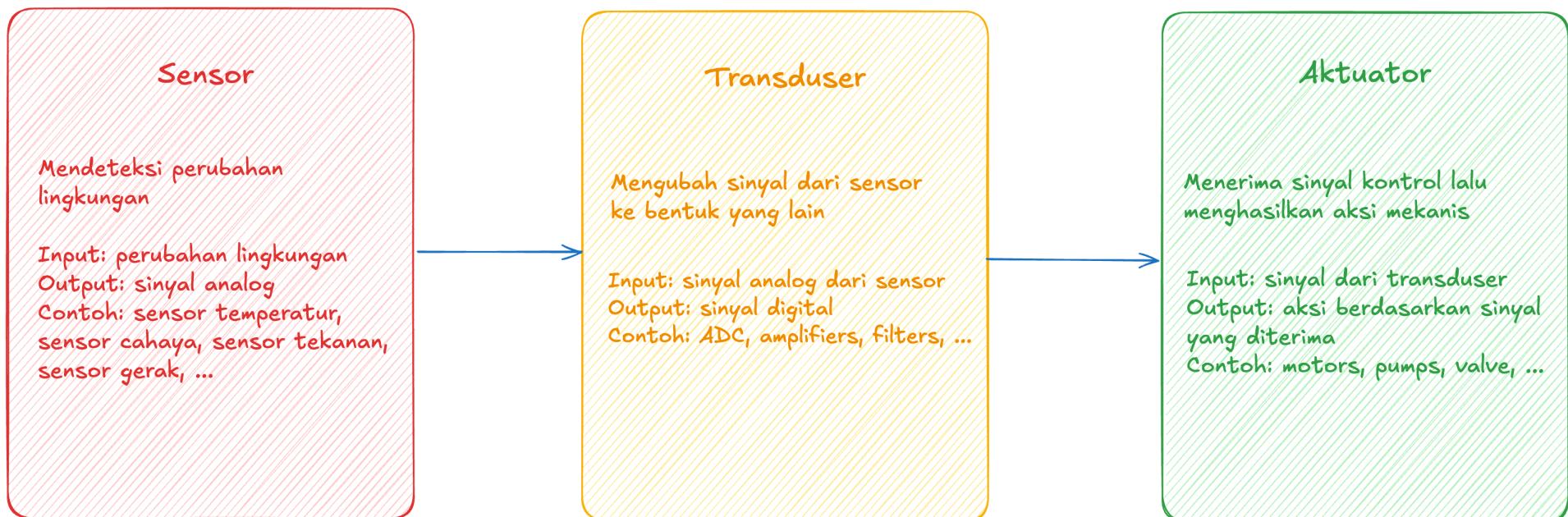


Fig. 2.1 Functional diagram of a basic data acquisition system

Sensor – Transduser – Aktuator



Pengondisian sinyal

Urgensi: sinyal sebelum diteruskan perangkat lain, misal ADC atau DAC harus diatur sedemikian rupa supaya **dapat meningkatkan performa dan kualitas sistem serta mencegah kerusakan perangkat**

Contoh: amplifikasi, pergeseran tegangan, filtering

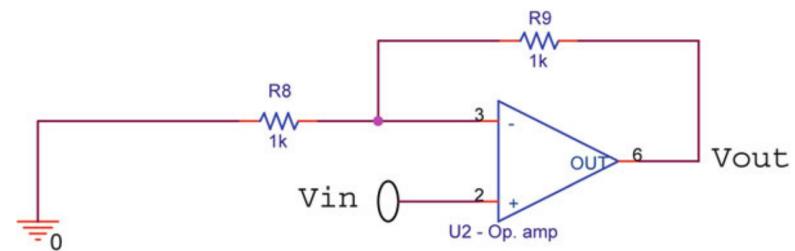


Fig. 2.4 Signal conditioning: amplification

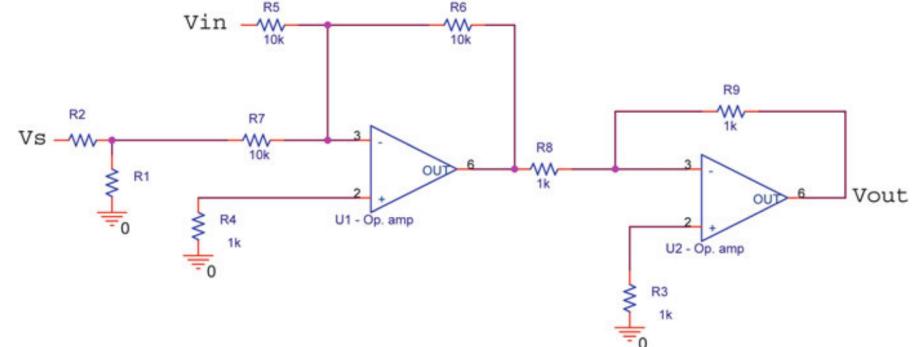


Fig. 2.5 Signal conditioning: shifting voltage

Analog to Digital Converter

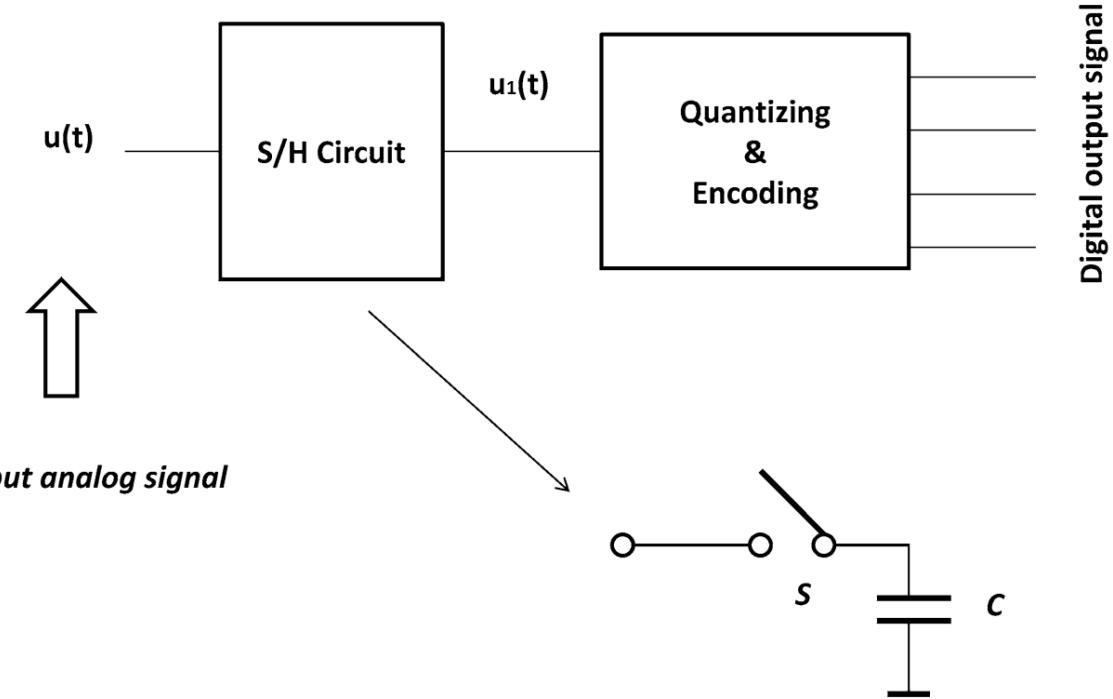


Fig. 2.10 Outline of analog to digital converter

Digital Signal Processing

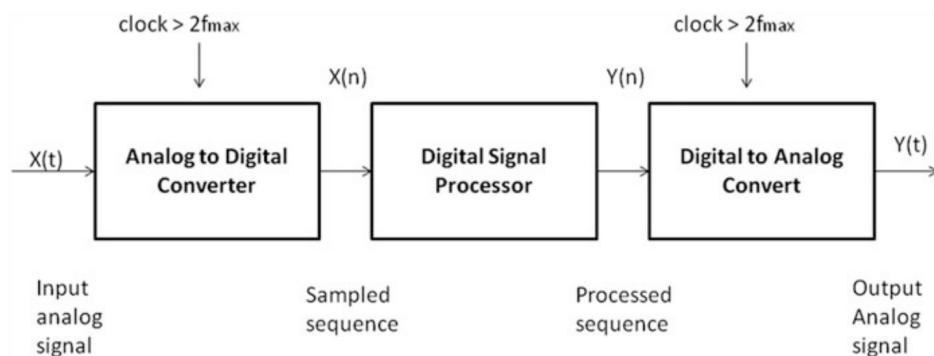


Fig. 2.43 Digital signal processing

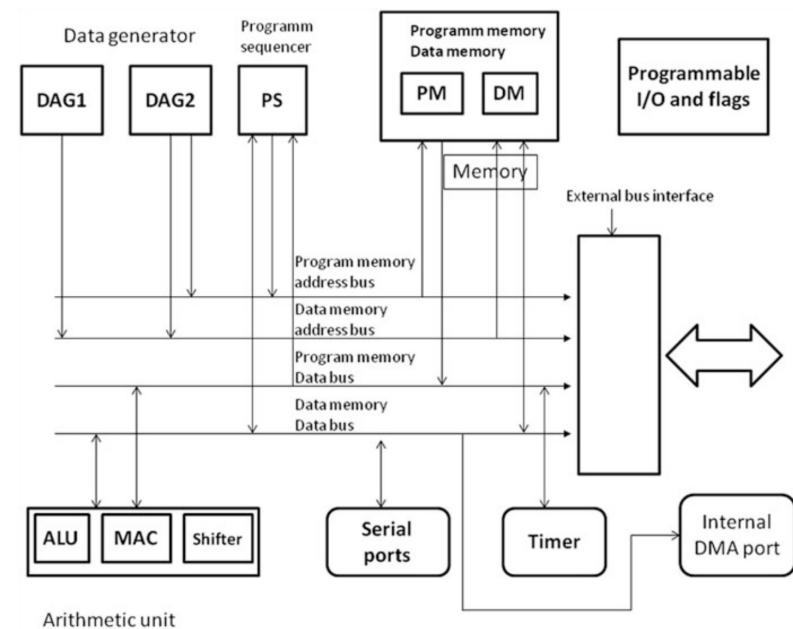


Fig. 2.44 Architecture of a DSP

Digital to Analog Converter

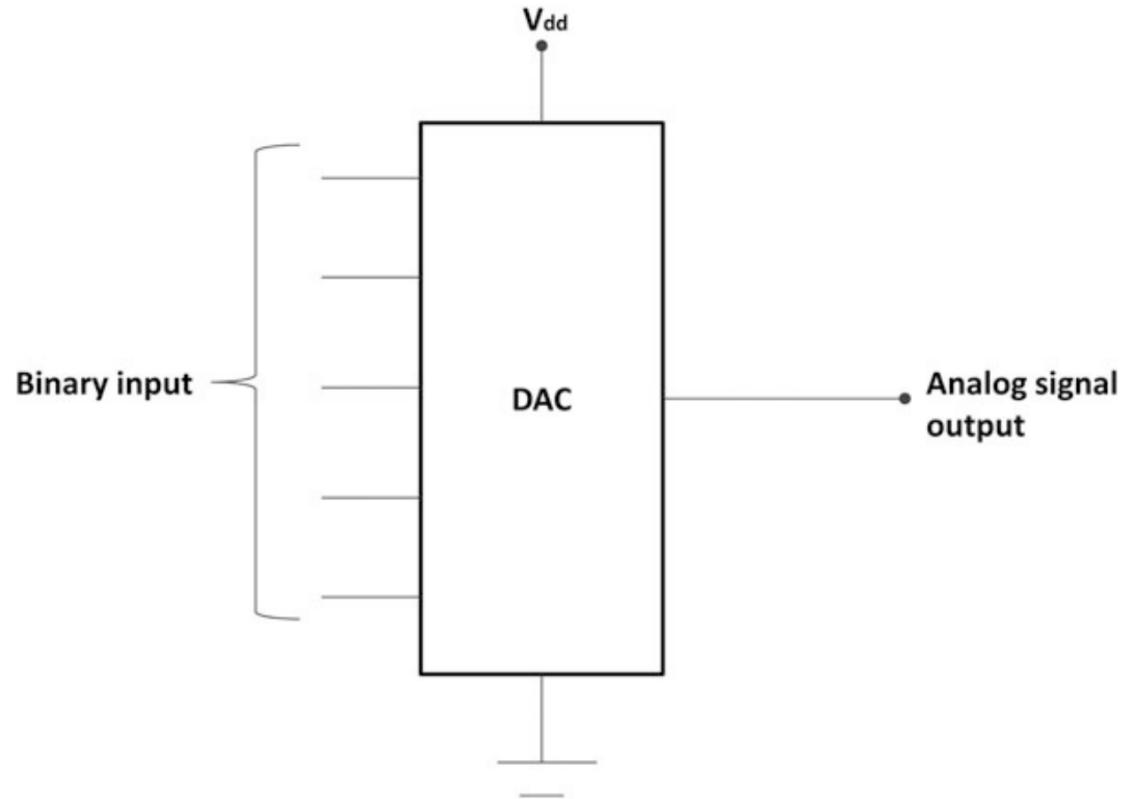


Fig. 2.28 Converter D/A

Communication bus

*Bus adalah **sistem komunikasi** yang digunakan untuk **transfer data antara komponen satu dengan yang lain** didalam rangkaian elektronik atau computer.*

Contoh: USB, firewire, RS-232 serial interface, PCI/PCIe, dan lain sebagainya

Prinsip perancangan perangkat lunak DAQ

Perangkat lunak pada DAQ harus dirancang untuk dapat melakukan *recovery* dengan baik jika ada komponen yang tidak berfungsi dengan baik ataupun kehilangan pasokan listrik tanpa kehilangan data.

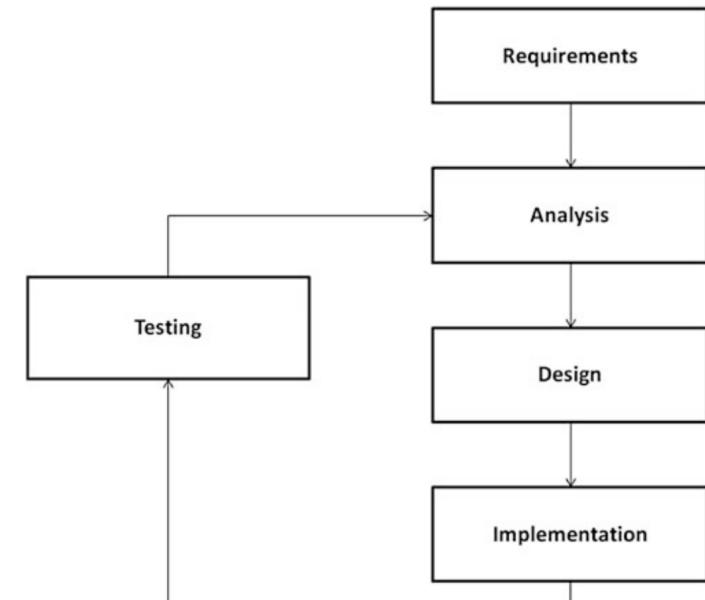


Fig. 5.1 Design software

Kebutuhan fungsional perangkat lunak DAQ

 Real-time monitoring

 Data analysis

 Data logging

 Control algorithms

 Human-machine-interface

Diskusi?



DIKTISAINTEK
BERDAMPAK

Pertemuan 3

Mikrokontroler

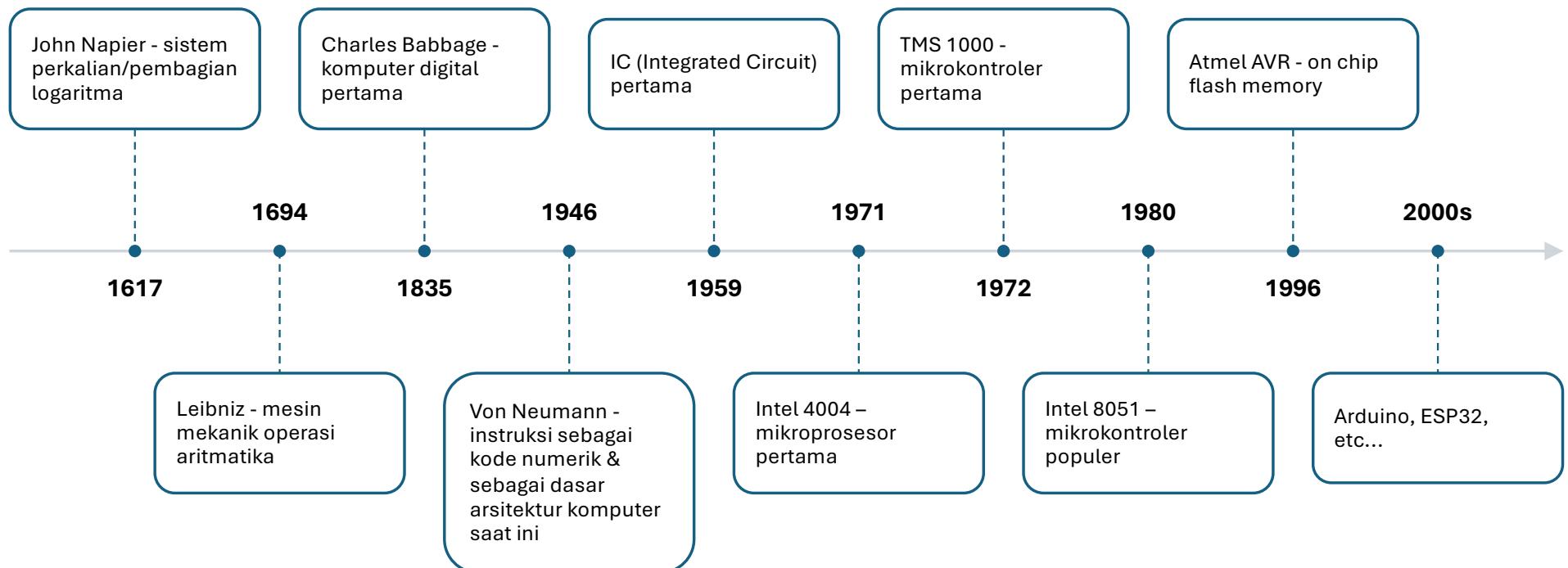
Ali Idrus, S.Kom., M.Kom.

Muhammad Ridho Kurniawan Pratama, S.Kom., M.T.I.

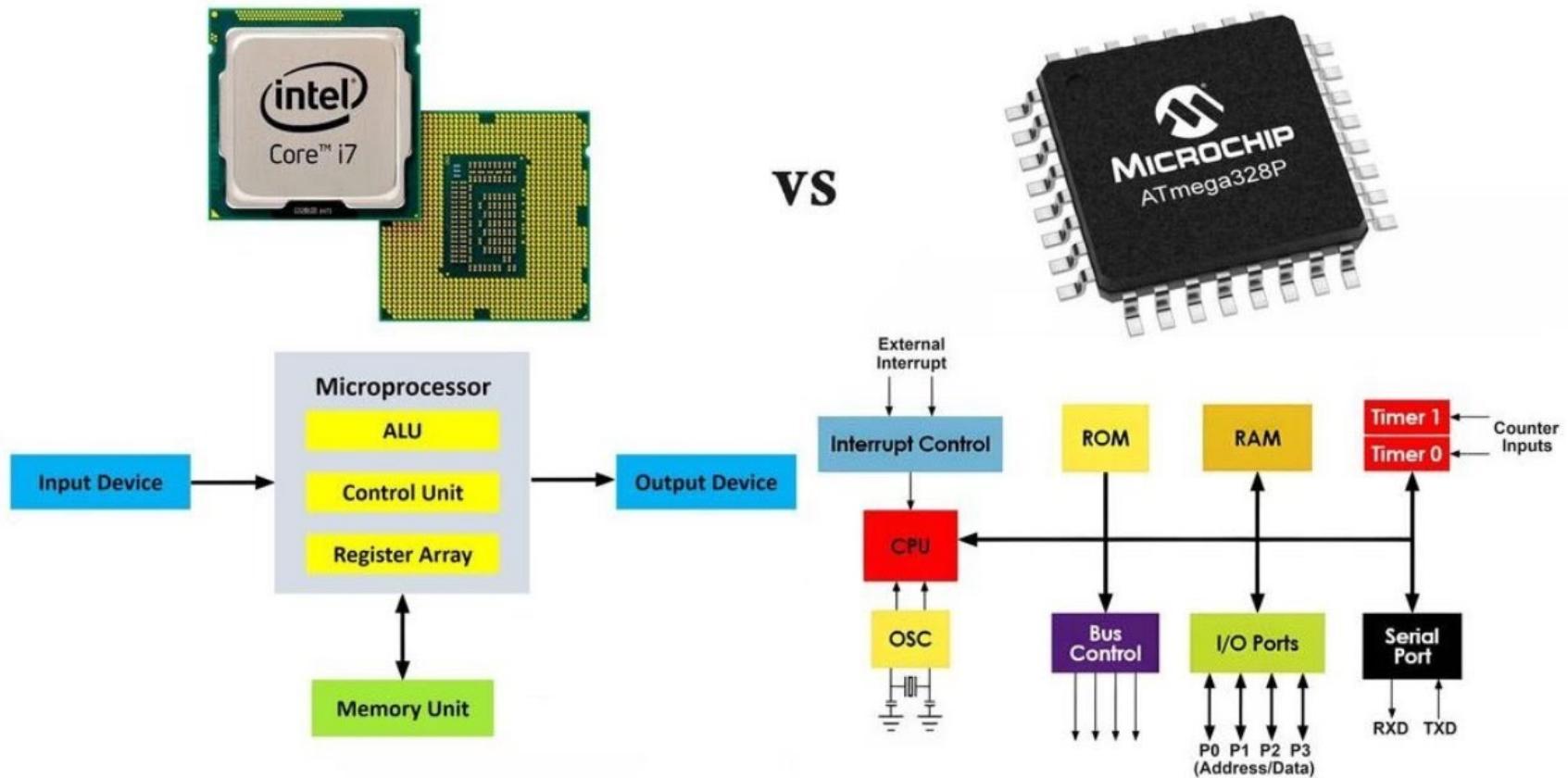
Tujuan Pembelajaran

-  17 Menjelaskan sejarah mikrokontroler
-  Membedakan mikrokontroler dan mikroprosesor
-  Mengetahui perbedaan arsitektur CISC dan RISC
-  Mengetahui berbagai matriks perbandingan antar mikrokontroler yang ada di pasaran (AVR, Arduino, ESP32)
-  Mengetahui pola dalam pemrograman mikrokontroler, khususnya pada Arduino

Sejarah Mikrokontroler



Mikroprocessor vs Mikrokontroler



CISC vs RISC

| Aspect | RISC | CISC |
|---------------------|--|---|
| Instruction Set | Small, simple, fixed-size | Large, complex, variable-size |
| Clock Cycles | 1 cycle per instruction | Multiple cycles per instruction |
| Decoding | Simple and fast | Complex and slow |
| Pipeline | Excellent pipelining support | Difficult to pipeline |
| Power Consumption | Lower power consumption - ideal for portable devices | Higher power consumption due to complex hardware |
| Code Size | Larger code size (more instructions needed) | Smaller code size - complex operations in single instructions |
| Memory Usage | More RAM required | Memory efficient |
| Hardware Complexity | Simpler architecture, fewer transistors | More complex circuitry, challenging to design |
| Execution Speed | Much faster execution rate than CISC | Slower execution speed due to complex instructions |
| Implementation | ARM-based microprocessor | x86-based microprocessor |

Komparasi mikrokontroler

| Platform | Type | Architecture | Core | Popular Models | Key Strength |
|----------|-----------------|----------------------------|-------------|---------------------------|-----------------------|
| AVR | Microcontroller | 8-bit RISC Harvard | Single | ATmega328P, ATmega2560 | Ultra-low power |
| Arduino | Dev Platform | 8-bit AVR / 32-bit ARM | Single/Dual | Uno R3, Uno R4, Nano | Ease of use |
| ESP32 | Microcontroller | 32-bit Xtensa LX6/LX7 RISC | Dual | ESP32, ESP32-S3, ESP32-C3 | Built-in connectivity |

| Specification | AVR (ATmega328P) | Arduino (Uno R3/R4) | ESP32 |
|--------------------------|------------------|---------------------------|-------------------|
| Flash Memory | 32 KB | 32KB (R3) / 256KB (R4) | 448KB - 16MB |
| SRAM | 2 KB | 2KB (R3) / 32KB (R4) | 320KB - 520KB |
| EEPROM | 1 KB | 1KB (R3) / 8KB (R4) | None (uses Flash) |
| Clock Speed | 16 MHz | 16MHz (R3) / 48MHz (R4) | Up to 240 MHz |
| Operating Voltage | 1.8V - 5.5V | 5V | 3.3V |
| GPIO Pins | 23 | 14-20 | 34-48 |
| ADC Resolution | 10-bit | 10-bit (R3) / 14-bit (R4) | 12-bit |

| Feature | AVR | Arduino | ESP32 |
|------------------------------------|-------------------|--|---------------------------------------|
| Wi-Fi | ✗ External module | ✗ Shield required (R4 WiFi has built-in) | ✓ Built-in 802.11 b/g/n |
| Bluetooth | ✗ External module | ✗ Shield required (R4 WiFi has BLE 5.0) | ✓ BLE & Classic |
| USB | ✗ UART only | ✓ USB-C (R4) / Micro-USB (R3) | ✓ Via CP2102/CH340 |
| Real-time OS | ✗ Bare metal | ✗ Bare metal | ✓ FreeRTOS |
| Floating Point | ✗ Software only | Hardware FPU (R4 only) | ✓ Hardware FPU |
| Controller Area Network Bus | ✗ | ✓ (R4 only) | ✓ TWAI (Two way automotive interface) |

| Metric | AVR (ATmega328P) | Arduino R4 | ESP32 |
|------------------------------|---------------------------|-----------------|-----------------------------------|
| Processing Power | ~1 MIPS/MHz | ~1.25 DMIPS/MHz | up to 600 DMIPS |
| Power Consumption | 1.5mA active, nA sleep | 10-20mA active | 20-240mA active, 2.5µA deep sleep |
| Battery Life | Years | Months | Days to Months |
| Real-time Performance | Excellent (deterministic) | Very Good | Good (RTOS) |
| Memory Bandwidth | Limited | Enhanced | High |
| Multitasking | Cooperative only | Enhanced (ARM) | Preemptive (dual-core) |

| Component Cost | AVR | Arduino | ESP32 |
|----------------------------|---------------------|-----------------|------------|
| Chip Cost | \$1 - \$3 | N/A | \$2 - \$6 |
| Dev Board | \$3 - \$10 | \$15-35 (R3-R4) | \$5 - \$15 |
| External Components | Crystal, USB-serial | Minimal | Minimal |
| Project Total | Very Low | Low-Moderate | Low |

| Use Case | AVR Rating | Arduino Rating | ESP32 Rating | Best Choice |
|--------------------|------------|----------------|--------------|-------------|
| Education/Learning | ★★★ | ★★★★★ | ★★★★★ | Arduino R4 |
| IoT/Smart Home | ★★ | ★★★ (R4 WiFi) | ★★★★★ | ESP32 |
| Battery-powered | ★★★★★ | ★★★★★ | ★★★★ | AVR |
| Real-time Control | ★★★★★ | ★★★★★ | ★★★★★ | AVR |
| Rapid Prototyping | ★★★ | ★★★★★ | ★★★★★ | Arduino |
| AI/ML Applications | ★ | ★★ | ★★★★★ | ESP32-S3 |
| Audio Processing | ★★ | ★★★ (R4 DAC) | ★★★★★ | ESP32 |

| Development Aspect | AVR | Arduino | ESP32 |
|--------------------|------------------|-------------------------|--------------------------|
| Primary IDE | Microchip Studio | Arduino IDE 2.0 | ESP-IDF / Arduino IDE |
| Programming | C/C++, Assembly | C++ (simplified) | C/C++, MicroPython, Rust |
| Learning Curve | ★★★★★ Advanced | ★★ Beginner | ★★★★★ Intermediate |
| Community | ★★★ Mature | ★★★★★ Huge | ★★★★★ Growing |
| Libraries | ★★★ Stable | ★★★★★ Extensive | ★★★★★ Comprehensive |
| Documentation | ★★★★★ Technical | ★★★★★ Beginner-friendly | ★★★★★ Professional |

Panduan pemilihan mikrokontroler

| AVR | Arduino | ESP32 |
|-------------------------------------|-------------------------------|------------------------------|
| • Deterministic real-time | • Learning/teaching | • IoT connectivity essential |
| • Ultra-low power (μA) | • Shield compatibility needed | • Dual-core processing |
| • Simple, dedicated tasks | • Large community support | • Built-in wireless |
| • Maximum battery life | • Rapid prototyping | • Advanced peripherals |
| • Cost under \$5 total | • Beginner-friendly | • AI/ML capabilities |

“Hello world” pada Arduino-based microcontroller

```
int ledPin = 9;      // the PWM pin the LED is attached to
int brightness = 0;  // how bright the LED is
int fadeAmount = 5;  // how many points to fade the LED by

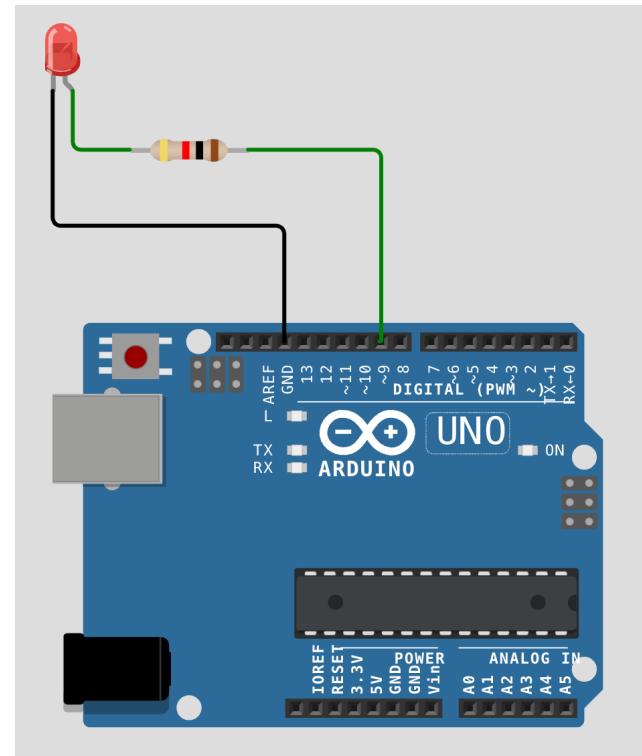
void setup() {
  // put your setup code here, to run once:
  pinMode(ledPin, OUTPUT);
}

void loop() {
  analogWrite(ledPin, brightness);

  // change the brightness for next time through the loop:
  brightness = brightness + fadeAmount;

  // reverse the direction of the fading at the ends of the fade:
  if (brightness <= 0 || brightness >= 255) {
    fadeAmount = -fadeAmount;
  }

  // wait for 50 milliseconds to see the dimming effect
  delay(50);
}
```



Referensi Tambahan

- **Microchip Technology** (2024). "ATmega328P Datasheet" - Official specifications
- **Arduino Store** (2024). "Arduino UNO R4 Minima" - RA4M1 specifications
- **Arduino Store** (2024). "Arduino UNO R4 WiFi" - Dual MCU architecture
- **Espressif Systems** (2024). "ESP32 Series Datasheet Version 5.0"
- **Espressif Systems** (2024). "ESP32-S3 Series Datasheet Version 2.0"
- **Wikipedia** (2024). "ATmega328" - Current technical specifications
- **Components101** (2024). "ATmega328P Pinout, Features & Datasheet"
- **Renesas Electronics** (2024). "Arduino UNO R4 32-bit MCU Prototyping Board"
- **Arduino Blog** (2023). "Arduino UNO R4 is a giant leap forward"
- **Instructables** (2024). "Arduino Comparison: UNO R3, R4 Minima and R4 WiFi"



DIKTISAINTEK
BERDAMPAK

Pertemuan 5

Antarmuka dan Komunikasi Data

Ali Idrus, S.Kom., M.Kom.

Muhammad Ridho Kurniawan Pratama, S.Kom., M.T.I.

Tujuan pembelajaran

- Mampu menjelaskan prinsip kerja dan karakteristik antarmuka komunikasi data dasar (UART, SPI, I2C, CAN, Ethernet, USB, Bluetooth, ZigBee) dalam sistem embedded.
- Mampu mengklasifikasikan dan membandingkan antarmuka komunikasi berdasarkan kecepatan, konsumsi daya, keandalan, jarak, dan kompleksitas untuk aplikasi spesifik.
- Mampu menjelaskan arsitektur hardware/software sistem komunikasi termasuk komponen perangkat keras, protocol stack, dan framework berbasis komponen (CCPA, IRES).
- Mampu memilih dan merekomendasikan antarmuka komunikasi yang tepat untuk aplikasi tertentu dengan justifikasi teknis berdasarkan literatur ilmiah.

Apa itu antar muka komunikasi data?

- **Definisi:** Mekanisme perangkat keras dan perangkat lunak yang memungkinkan transfer data antar perangkat dalam sistem embedded
- **Fungsi Inti:** Menghubungkan sensor, aktuator, kontroler, dan periferal
- **Persyaratan Utama:** Keandalan, efisiensi, dan kecepatan yang sesuai untuk aplikasi
- **Area Dampak:** Performa sistem, konsumsi energi, dan biaya

Antarmuka komunikasi serial umum

UART (Universal Asynchronous Receiver/Transmitter):

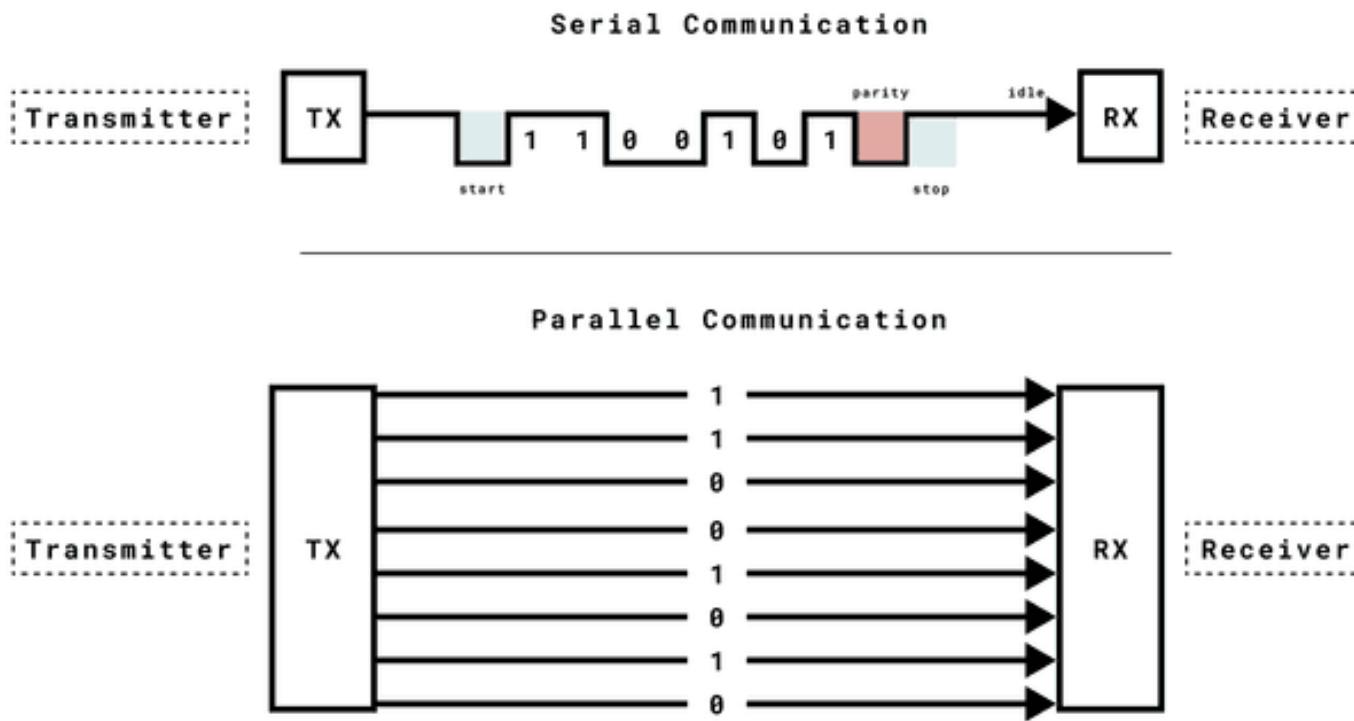
- Komunikasi serial point-to-point yang sederhana dan berkecepatan rendah
- Banyak digunakan untuk debugging, interfacing sensor, dan dukungan perangkat lawas (*legacy*)

SPI (Serial Peripheral Interface):

- Protokol master-slave full-duplex berkecepatan tinggi
- Ideal untuk komunikasi throughput tinggi jarak pendek dengan sensor, memori, dan display

I2C (Inter-Integrated Circuit):

- Protokol multi-master half-duplex dua jalur
- Cocok untuk komunikasi kecepatan sedang dan daya rendah dengan beberapa periferal



Protokol komunikasi yang lain

CAN (Controller Area Network)

- Protokol berbasis pesan multi-master yang robust
- Standar dalam otomotif dan kontrol industri untuk komunikasi real-time yang fault-tolerant

Ethernet (termasuk TSN)

- Berkecepatan tinggi, scalable, semakin deterministik dengan TSN
- Digunakan dalam otomasi industri, otomotif, dan sistem embedded berperforma tinggi

USB (Universal Serial Bus)

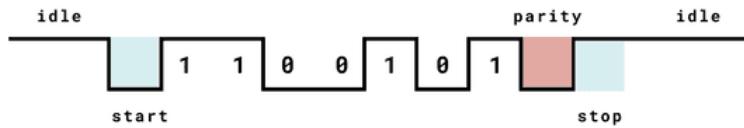
- Antarmuka plug-and-play yang fleksibel
- Mendukung berbagai perangkat dari antarmuka manusia hingga penyimpanan dan media

Wireless (Bluetooth, ZigBee):

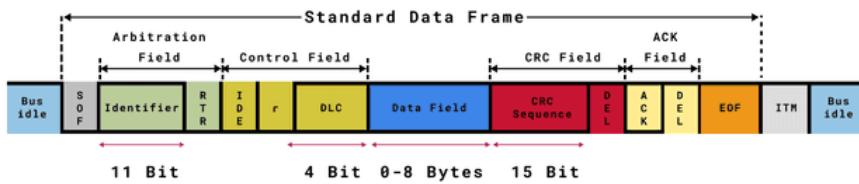
- Bluetooth: Kecepatan data lebih tinggi, jarak pendek
- ZigBee: Jaringan sensor berdaya rendah

Bagaimana format data yang dikirimkan?

UART Frame Format



CAN Frame Format



I2C Message



Gambar: arduino.cc

Mengapa serial lebih umum digunakan dibanding komunikasi paralel?

Beberapa masalah pada komunikasi paralel

- Jalur data yang lebih dari satu dapat memiliki delay propagasi yang berbeda-beda, lebih sulit dalam melakukan sinkronisasi data ketika dalam *clock frequency* tinggi (*skew issue*)
- Adanya resiko interferensi antar jalur data yang berdekatan
- Rangkaian elektronika yang lebih kompleks
- Batasan jarak antara komponen

Keuntungan pada komunikasi serial

- Jalur data yang lebih sedikit, rangkaian elektronikanya menjadi lebih sederhana, dan lebih hemat daya
- Dapat melakukan transfer data dengan *clock frequency* tinggi tanpa adanya *skew issue*
- Interferensi yang dihasilkan lebih minimum
- Dapat mencapai jarak yang lebih jauh bergantung protokol yang digunakan (misal ethernet)

Klasifikasi antar muka

| | |
|-----------------------------|---|
| Kecepatan | Kecepatan rendah: UART, I2C Kecepatan tinggi: Ethernet, USB, SPI |
| Konsumsi Daya | Daya rendah: I2C, ZigBee Daya lebih tinggi: Ethernet, USB |
| Keandalan | Fault-tolerant: CAN, TSN (Time-sensitive Networking) Ethernet Best-effort: UART, SPI |
| Jarak | Jarak pendek: SPI, I2C Menengah: CAN, RS422 Jarak jauh: Ethernet, wireless |
| Persyaratan Aplikasi | Real-time, deterministik: CAN, TSN Fleksibel: USB, Bluetooth Scalable: Ethernet |

Memilih antarmuka yang sesuai

Penggunaan spesifik aplikasi

- **Sensor local/komunikasi periferal:** UART, SPI, I2C
- **Otomasi dan kontrol waktu nyata:** CAN, Ethernet (TSN)
- **Transfer data berkecepatan tinggi:** Ethernet, USB
- **Wireless sensor networks:** Bluetooth, ZigBee

Faktor pemilihan

- Besaran data rate yang dibutuhkan
- Jumlah perangkat yang terhubung
- Batasan daya
- Jarak antar perangkat
- Kebutuhan akan kehandalan
- Toleransi pada kompleksitas sistem

Model konsumsi energi

Model konsumsi energi pada I2C

- Desain open-drain menyebabkan kehilangan energi statis untuk keadaan logic-0 (voltase mendekati 0V)
- Teknik encoding (misalnya, ACME) dapat mengurangi energi hingga 57%

Model konsumsi energi pada SPI

- Energi bergantung pada kecepatan clock, kecepatan data, dan tegangan supply
- Kecepatan lebih tinggi meningkatkan konsumsi daya
- Dapat dioptimalkan untuk beban kerja spesifik

[8,9,10,17,18]

Pendekatan Pemodelan

- Model tingkat instruksi dan tingkat sistem dengan memanfaatkan *overhead* perangkat keras dan OS
- Dilakukan untuk mencapai kesalahan estimasi di bawah 10%

Faktor Lingkungan

- Frekuensi prosesor mempengaruhi konsumsi
- Algoritma *scheduling* berdampak pada efisiensi
- Parameter komunikasi secara signifikan mempengaruhi penggunaan energi

Komponen perangkat keras

Mikroprosesor dan SoC

- ARM, DSP, Xilinx SoC, dan mikrokontroler lainnya
- Menyediakan tulang punggung komputasi
- Mengintegrasikan periferal komunikasi

IC Periferal

- Chip MPCC, DUART
- Mengimplementasikan protokol komunikasi spesifik (UART, SPI)

Prosesor I/O dan Bus Bridge

- PCI dan local bus bridge
- Kontroler I/O khusus
- Mengelola aliran data antar komponen sistem

Arsitektur & protokol perangkat lunak

Arsitektur Berbasis Komponen

CCPA (Component-Based Communication Protocol Architecture)

- Memodularisasi protocol stack
- Menyediakan fleksibilitas, efisiensi memori, dan konkurensi

IRES Framework

- Mendukung rekonfigurasi dinamis
- Memungkinkan adaptasi runtime yang aman

Protocol Stack

- Implementasi TCP/IP, UDP, SPI, RS232, 1-Wire, stack wireless khusus
- Sering terintegrasi dengan OS (misalnya, Linux)

Verifikasi

- SystemVerilog dan UVM untuk verifikasi protokol
- Metode formal memastikan *correctness* dan *robustness*

Matriks perbandingan antar muka

| Antarmuka | Kecepatan | Keandalan | Daya | Kompleksitas | Penggunaan |
|-----------|-----------|--------------|---------------|--------------|--------------------------|
| UART | Rendah | Sedang | Rendah | Rendah | Debug, sensor |
| SPI | Tinggi | Tinggi | Sedang | Sedang | Memori, display |
| I2C | Sedang | Sedang | Rendah | Rendah | Sensor, EEPROM |
| CAN | Sedang | Tinggi | Rendah | Sedang | Otomotif, industri |
| Ethernet | Tinggi | Tinggi (TSN) | Tinggi | Tinggi | Industri, real-time |
| USB | Tinggi | Tinggi | Tinggi | Tinggi | Periferal, penyimpanan |
| Bluetooth | Sedang | Sedang | Rendah | Sedang | Periferal nirkabel |
| ZigBee | Rendah | Sedang | Sangat Rendah | Rendah | Jaringan sensor nirkabel |

[1,2,3]

Ringkasan

- Berbagai antarmuka (UART, SPI, I2C, CAN, Ethernet, USB, wireless) diaplikasikan pada beberapa kasus berdasarkan persyaratan kecepatan, daya, keandalan, jarak, dan kompleksitas.
- Pemilihan antarmuka memerlukan penyeimbangan batasan aplikasi termasuk kecepatan data, jumlah perangkat, budget daya, jarak, dan kebutuhan keandalan.
- Desain sistem yang sukses mengintegrasikan komponen hardware, framework software, dan *trade-off* kinerja untuk memenuhi persyaratan aplikasi embedded yang spesifik.

Referensi

- [1] Minakov, I., Rojas, S. (2024). Performance Analysis of a Communication Interface for Control and Data Visualization of Power Converters. ECCE Europe 2024.
- [2] Armstrong, J., Helps, C.R. (2007). Comparative evaluation of ZigBee and bluetooth: Embedded wireless network technologies for students and designers. ASEE Annual Conference.
- [3] Dragusin, S.-A., Bizon, N., Teodorescu, R.-M., Anghel, G. (2025). Communication Protocols in Embedded Systems for Automotive Applications. ECAI 2025.
- [4] Dai, H.-J., Chen, T.-Z., Chen, C. (2005). CCPA: Component-based communication protocol architecture for embedded systems. Journal of Zhejiang University: Science.
- [5] Matthys, N., Hughes, D., Michiels, S., Joosen, W. (2009). Fine-grained tailoring of component behaviour for embedded systems. Lecture Notes in Computer Science.
- [6] Nguyen, H.K., Tran, X.-T. (2016). Design and implementation of a hybrid switching router for the reconfigurable network-on-chip. International Conference on Advanced Technologies for Communications.
- [7] Lusala, A.K., Legat, J.-D. (2012). Combining SDM-based circuit switching with packet switching in a router for on-chip networks. International Journal of Reconfigurable Computing.
- [8] Xie, C., Pagliari, D.J., Calimera, A., Poncino, M. (2021). ACME: An Energy-Efficient Approximate Bus Encoding for I²C. Proceedings of the International Symposium on Low Power Electronics and Design.
- [9] Dhouib, S., Senn, E., Diguet, J.-P., Laurent, J. (2009). Energy and power consumption estimation for embedded applications and operating systems. Journal of Low Power Electronics.
- [10] Kumar, A., Yadav, S.K., Pareek, R., Gour, N. (2023). Analysis of discrete voltage level energy efficient scheduling for fixed priority framework. International Journal of Wireless and Mobile Computing.
- [11] Wang, F., Li, X., Hu, S., Wang, J. (2016). Remote communication interface of embedded system based on 10Gigabit Ethernet. ICSPCC 2016.
- [12] Zhou, L., Song, N., Feng, Y., Duan, H. (2023). Design of embedded software with RS422 interface for highly reliable data reception. ICBASE 2023.
- [13] Daraban, M., Corches, C., Taut, A., Chindris, G. (2021). Protocol over UART for Real-Time Applications. SIITME 2021.
- [14] Mittag, L. (2005). Software unsnarls embedded connections. Electronic Engineering Times.
- [15] Wong, W. (2003). Interface options abound for MCU networking. Electronic Design.

Referensi (lanjutan)

- [16] Chandrashekhar, M.S., Kumbhare, A. (2023). The Integrated SDL-based design approach to create and implement wireless communication protocol. *Journal of Integrated Science and Technology*.
- [17] Jamaludin, I.I.B., Hassan, H.B. (2020). Design and Analysis of Serial Peripheral Interface for Automotive Controller. SCOReD 2020.
- [18] Kamuda, K., Klepacki, D., Sabat, W., Rzeszowska, P. (2015). Energetic properties of spi serial communication interface. *Przeglad Elektrotechniczny*.
- [19] Robinson, Robert (1998). Using I₂O and I/O processors in embedded PCI systems. *Wescon Conference Record*.
- [20] Magill, Jim, Thomsen, Guy (1984). COMPLEX DATACOMM PERIPHERAL ICs INTERFACE TO MANY PROCESSORS. EDN.
- [21] Cao, X.-L., Wu, P., Ding, T.-F. (2005). Design of a communication terminal based on ARM and DSP. *Journal of Electron Devices*.
- [22] Sharmila, K.P., Mukherjee, A. (2019). System verilog based verification for serial peripheral interface protocol to get maximum coverage. *Journal of Advanced Research in Dynamical and Control Systems*.
- [23] Chiu, J.-C., Yeh, T.-L. (2010). IRES: An integrated software and hardware interface framework for reconfigurable embedded system. *IET Computers and Digital Techniques*.
- [24] Tianzhou, C., Hongjun, D., Peng, Y., Wei, H. (2006). The assembly and maintenance of the component-based communication protocols for embedded system. *IEEE Conference on Industrial Electronics and Applications*.
- [25] Chen, T., Gan, Q., Hu, W., Huang, J. (2005). Communication protocol decomposition and component-based protocol submodule. *Proceedings of the IASTED International Conference on Software Engineering*.
- [26] Foukalas, F., Ntarladimas, Y., Glentis, A., Boufidis, Z. (2005). Protocol reconfiguration using component-based design. *Lecture Notes in Computer Science*.
- [27] Niamanesh, M., Jalili, R. (2007). A dynamic-reconfigurable architecture for protocol stacks of networked systems. *Proceedings - International Computer Software and Applications Conference*.
- [28] Niamanesh, M., Sabetghadam, S., Rahaghi, R.Y., Jalili, R. (2007). Design and implementation of a dynamic-reconfigurable architecture for protocol stack. *Lecture Notes in Computer Science*.
- [29] Pissias, P., Coulson, G. (2008). Framework for quiescence management in support of reconfigurable multi-threaded component-based systems. *IET Software*.
- [30] Kim, W.J., Lee, S.H., Hwang, S.Y. (2009). Design of an area-efficient and low-power hierarchical NoC architecture based on circuit switching. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*.
- [31] Jiang, G., Li, Z., Wang, F., Wei, S. (2015). A Low-Latency and Low-Power Hybrid Scheme for On-Chip Networks. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*.
- [32] Anders, M.A., Kaul, H., Krishnamurthy, R.K., Borkar, S.Y. (2011). Hybrid circuit/packet switched network for energy efficient on-chip interconnections. *Low Power Networks-On-Chip*.
- [33] Teimouri, N., Modarressi, M., Sarbazi-Azad, H. (2013). Power and performance efficient partial circuits in packet-switched networks-on-chip. *Proceedings of the 2013 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, PDP 2013*.



Pertemuan 4

Embedded Development Board

Ali Idrus, S.Kom., M.Kom.

Muhammad Ridho Kurniawan Pratama, S.Kom., M.T.I.

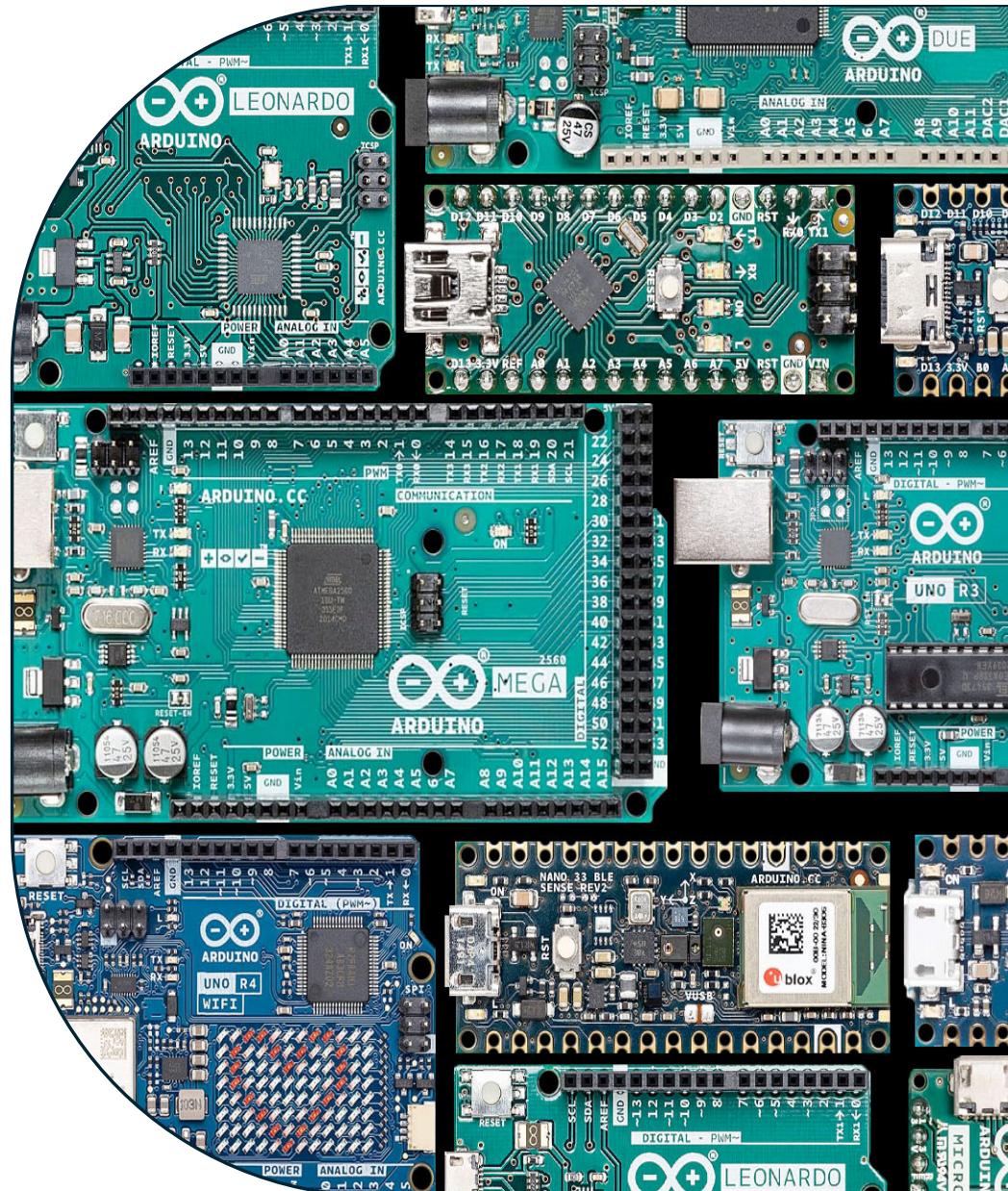
Tujuan Pembelajaran



- Mengetahui berbagai *embedded development board* seperti Arduino, Wemos dan Raspberry Pi
- Mengetahui perbandingan spesifikasi masing-masing *embedded development board* sesuai tipenya

Arduino

- Open source
- Dirancang di Ivrea Interaction Design untuk membuat prototype yang dikhususkan untuk siswa tanpa punya pengalaman elektronika dan pemrograman
- Ekosistem komprehensif, baik H/W, S/W, cloud platform ataupun perkakas pemrogramannya



| Model | Price | Processor | Clock | Memory | Connectivity | Key Features | Best For |
|----------------------|-------|---------------------------------------|-------------------|---|---------------------------|--------------------------|-----------------------------|
| UNO R4 Minima | \$20 | Renesas RA4M1 32-bit ARM Cortex-M4 | 48 MHz | 32KB SRAM 256 KB Flash | USB-C | 5V logic, Qwiic, CAN | Beginners, Education |
| UNO R4 WiFi | \$27 | RA4M1 + ESP32-S3 | 48 MHz + 240 MHz | 32KB + 512KB SRAM 256 KB + 384KB Flash | WiFi, Bluetooth, USB-C | LED Matrix, Dual MCU | IoT Projects |
| Nano ESP32 | \$18 | ESP32-S3 Dual-core | 240 MHz | 512KB SRAM 16MB Flash | WiFi, Bluetooth | Tiny size, Low power | Compact IoT |
| GIGA R1 WiFi | \$60 | STM32H747XI Dual Cortex-M7+M4 | 480 MHz + 240 MHz | 1MB SRAM 2MB Flash | WiFi, Bluetooth, Ethernet | High performance, AI/ML | Advanced Projects |
| Nano 33 BLE | \$22 | nRF52840 | 64 MHz | 256KB SRAM 1MB Flash | Bluetooth LE | Ultra-low power, sensors | Wearables |

Arduino Uno R4 WiFi

Summary

- 48 MHz Arm® Cortex®-M4 microprocessor with a floating point unit (FPU)
- 5 V operating voltage
- Real-time Clock (RTC)
- Memory Protection Unit (MPU)
- Digital-to-analog Converter (DAC)

Memory

- 256 kB Flash Memory
- 32 kB SRAM
- 8 kB Data Memory (EEPROM)

Arduino Uno R4 WiFi

Peripherals

- Capacitive Touch Sensing Unit (CTSU)
- USB 2.0 Full-Speed Module (USBFS)
- 14-bit ADC
- Up to 12-bit DAC
- Operational Amplifier (OPAMP)

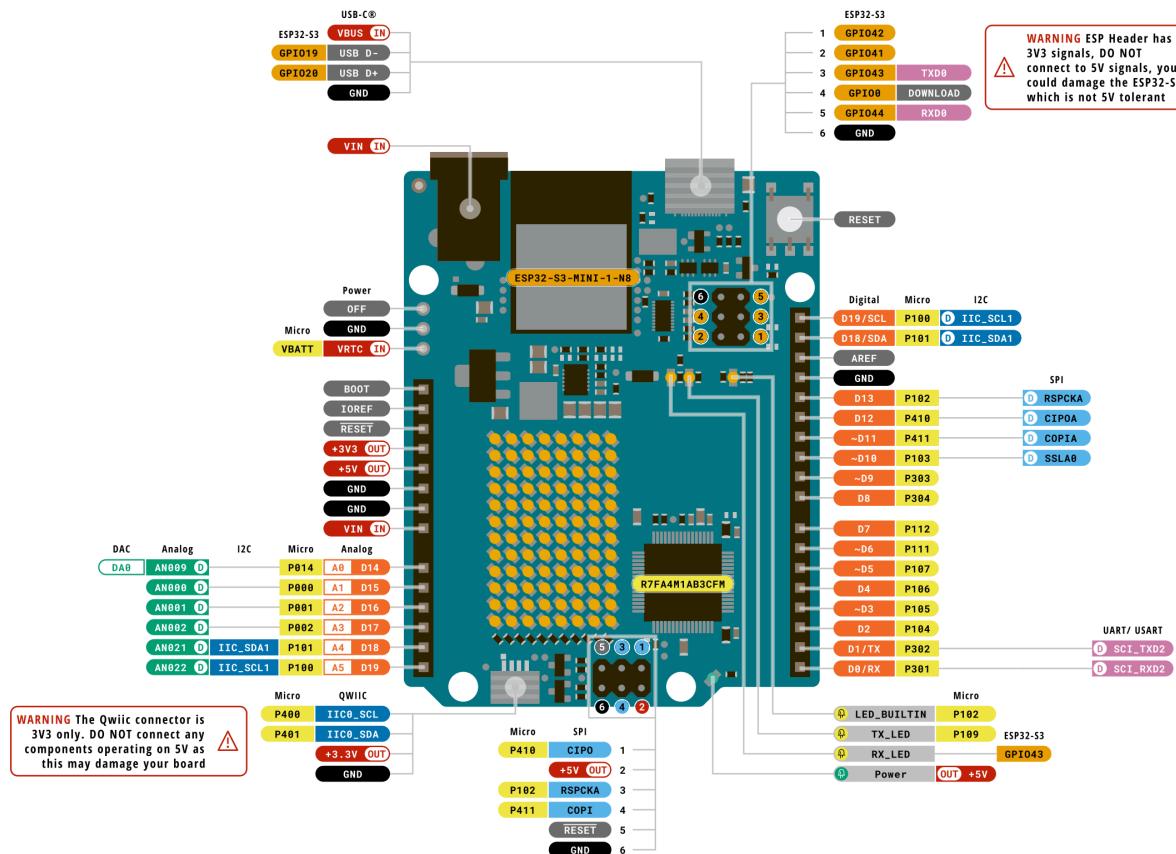
Power

- Operating voltage for RA4M1 is 5 V
- Recommended input voltage (VIN) is 6-24 V
- Barrel jack connected to VIN pin (6-24 V)
- Power via USB-C® at 5 V

Arduino Uno R4 WiFi

Communication

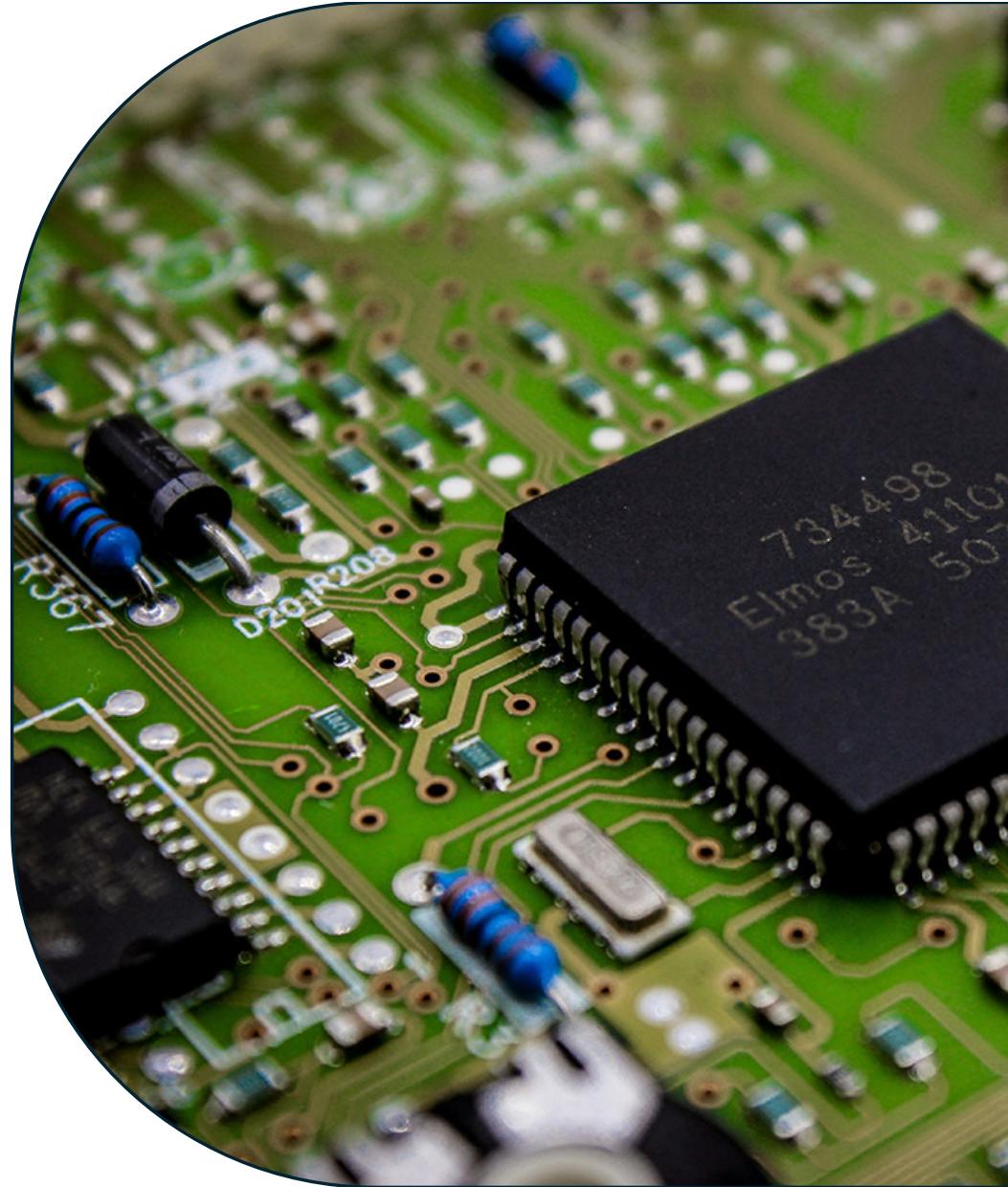
- 1x UART (pin D0, D1)
- 1x SPI (pin D10-D13, ICSP header)
- 2x I2C (pin A4, A5, SDA, SCL), (Qwiic connector)
- 1x CAN (pin D4, D5, external transceiver is required)



| Legend: | | Micro | | Micro | | Micro | |
|--------------------|---------------|---|-----------|---|-----------|-------|-----------|
| ■ Power | ■ Power Input | ■ GPIO Digital External | ■ Default | ■ LED | ■ Default | ■ LED | ■ Default |
| | | ■ Analog External | ■ Default | ■ RGB LED | ■ Default | | |
| ■ Ground | | ■ Main Part | ■ Default | | ■ Other | | |
| | | ■ Secondary Part | | | | | |
| | | ■ Internal Component | | | | | |
| | | ■ Other Pins (Reset, System Control, Debugging) | | | | | |
| | | | | i CPO/CPI have previously been referred to as MISO/MOSI | | | |
| ARDUINO | | Uno R4 WiFi Board | | | | | |
| | | | | | | | |
| SKU code: ABX00087 | | Full Pinout - Page 1 of 4 | | Last update: 23 Apr., 2025 | | | |
| DOCS.ARDUINO.CC | | | | This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/4.0/ or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA. | | | |
| | | | | | | | |

Wemos

- Sekarang berubah nama menjadi LOLIN
- Menawarkan konektivitas WiFi yang singkat
- Berbasis mikrokontroler ESP8266 dan ESP32



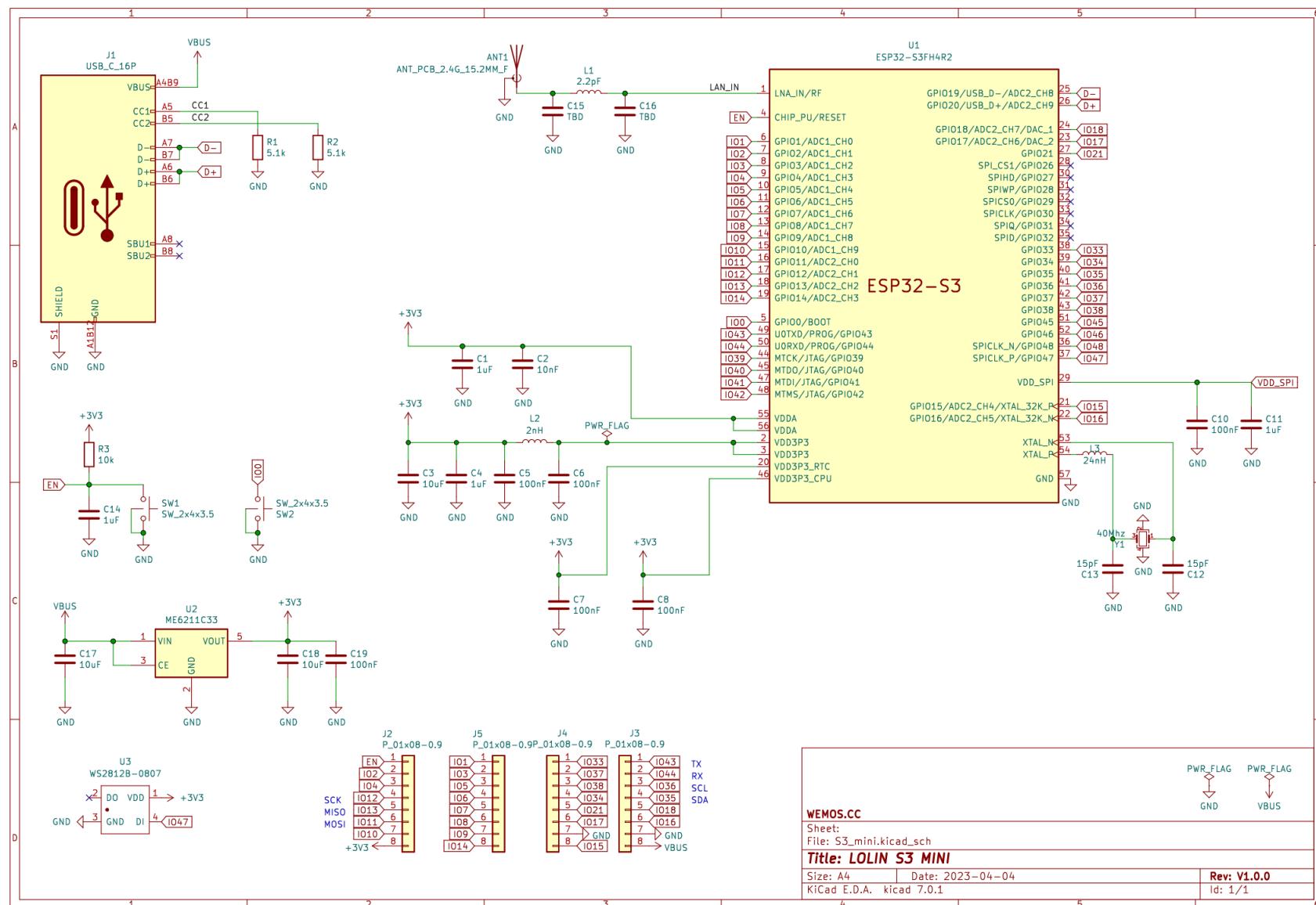
| Model | MCU | Clock | Memory | Size | WiFi | Bluetooth | Power | Price | Best For |
|---------------|----------|---------|------------------------|----------|-----------------------|-----------|-------|-------|-----------------|
| D1 Mini | ESP8266 | 80 MHz | 4MB Flash 80KB RAM | 34×25 mm | 2.4GHz | ✗ | 3.3V | ~\$3 | Budget IoT |
| D1 Mini Pro | ESP8266 | 80 MHz | 16MB Flash 80KB RAM | 34×25 mm | 2.4GHz + Ext. Antenna | ✗ | 3.3V | ~\$5 | Better Range |
| D1 Mini ESP32 | ESP32 | 240 MHz | 4MB Flash 520KB RAM | 40×30 mm | 2.4GHz | ✓ BLE 4.2 | 3.3V | ~\$6 | Dual-core Power |
| S2 Mini | ESP32-S2 | 240 MHz | 4MB Flash 320KB RAM | 34×25 mm | 2.4GHz | ✗ | 3.3V | ~\$5 | USB Native |
| S3 Mini | ESP32-S3 | 240 MHz | 8MB Flash 512KB RAM | 34×25 mm | 2.4GHz | ✓ BLE 5.0 | 3.3V | ~\$8 | Latest/Best |
| C3 Mini | ESP32-C3 | 160 MHz | 4MB Flash 400KB RAM | 25×20 mm | 2.4GHz | ✓ BLE 5.0 | 3.3V | ~\$4 | Compact RISC-V |

Wemos (LOLIN) S3 Mini

Features

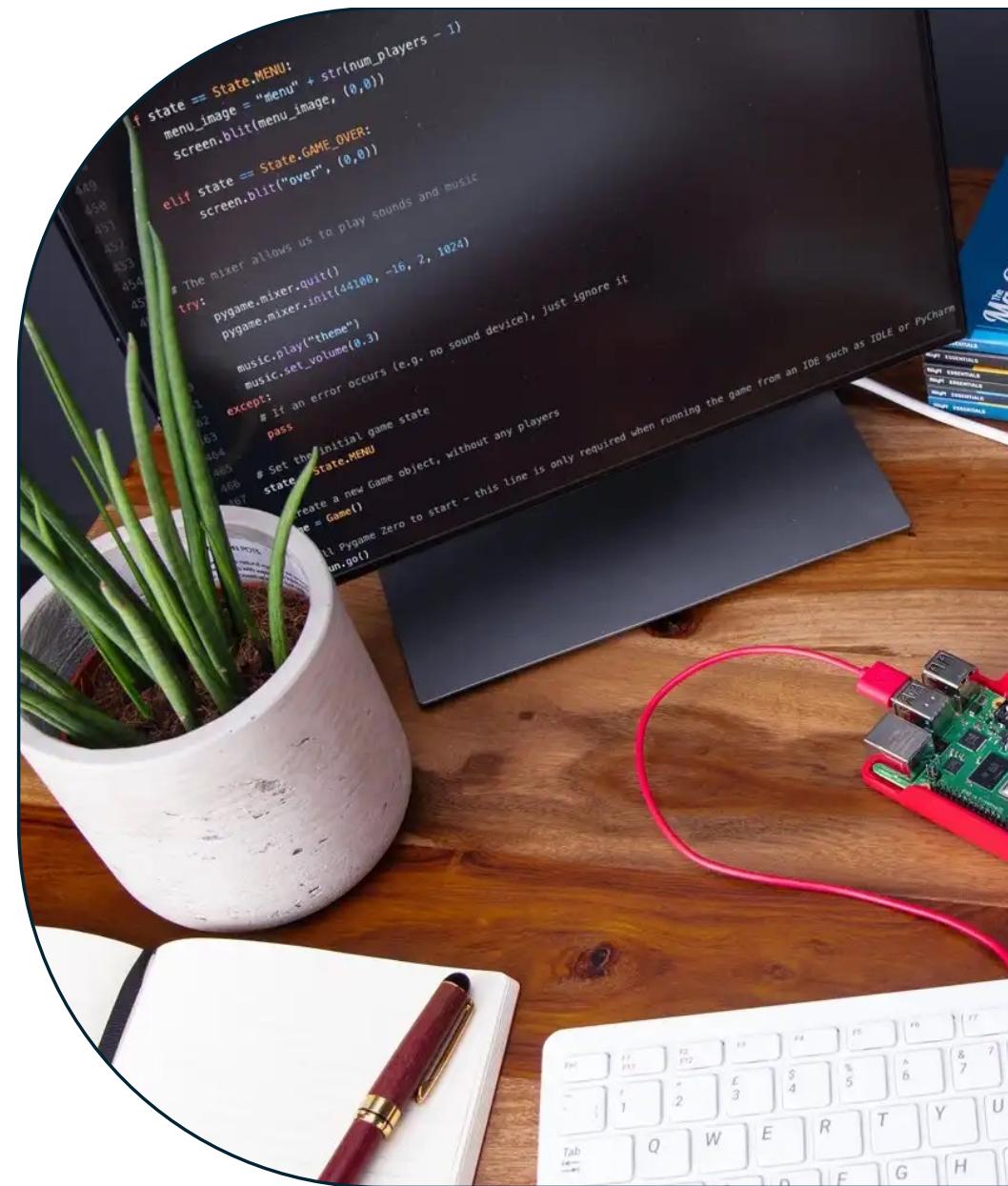
- based ESP32-S3FH4R2
- 2.4 GHz Wi-Fi
- Bluetooth LE
- 4MB Flash
- 2MB PSRAM
- 27x IO
- 1x RGB LED (IO47)
- ADC, DAC, I2C, SPI, UART, USB OTG
- Compatible with MicroPython, Arduino and ESP-IDF
- Default firmware: MicroPython

| | |
|--------------------------|-------------|
| Operating Voltage | 3.3V |
| Digital I/O Pins | 27 |
| Clock Speed | 240MHz |
| Flash | 4M Bytes |
| PSRAM | 2M Bytes |
| Size | 34.3*25.4mm |
| Weight | 3g |



Raspberry Pi

- Beberapa model sudah tergolong sebagai komputer (*single board computer*), bukan lagi mikrokontroler
- Menawarkan berbagai penggunaan yang lebih luas, dari sistem terbenam hingga komputer desktop

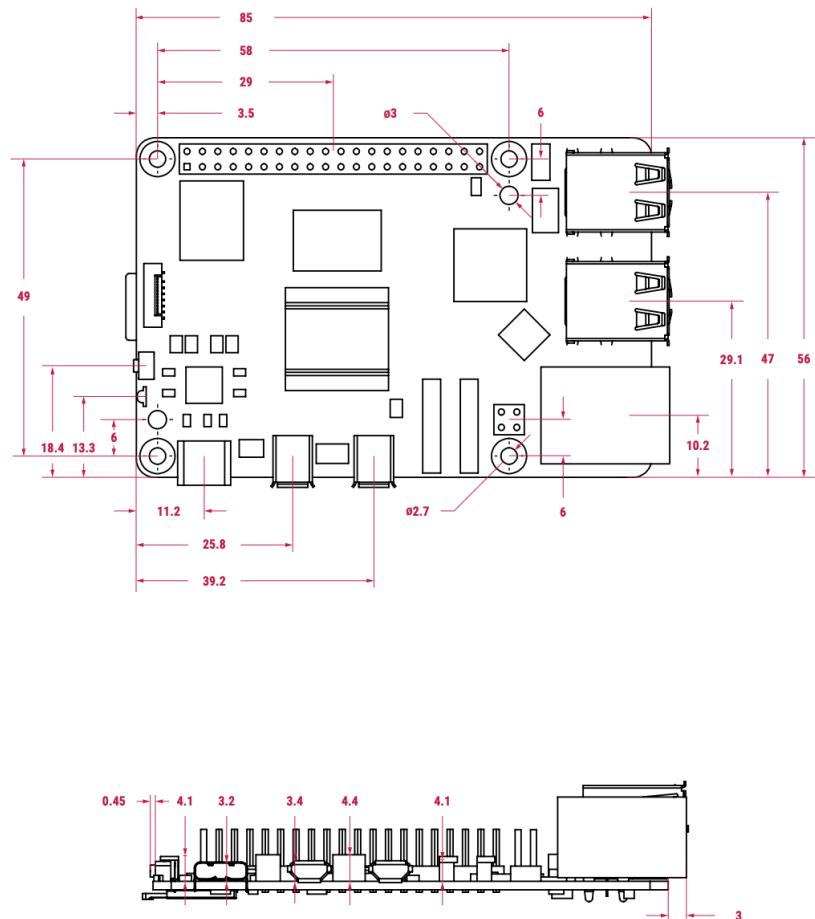


| Model | CPU | Clock | RAM Options | Storage | Price | Form Factor | Best For |
|-------------------|---------------------|--------------|--------------------|----------------|--------------|--------------------|------------------------|
| Pi 5 | BCM2712 Quad A76 | 2.4 GHz | 2/4/8/16GB | MicroSD + NVMe | \$60-\$150 | Standard | Desktop/Server |
| Pi 4B | BCM2711 Quad A72 | 1.8 GHz | 2/4/8GB | MicroSD | \$35-\$75 | Standard | General Purpose |
| Pi Zero 2W | RP3A0 Quad A53 | 1.0 GHz | 512MB | MicroSD | \$15 | Ultra-compact | IoT/Embedded |
| Pi 500 | BCM2712 Quad A76 | 2.4 GHz | 4GB | MicroSD | \$90 | Keyboard | All-in-one PC |
| Pico 2 | RP2350 Dual M33 | 150 MHz | 520KB | 4MB Flash | \$5 | Tiny MCU | Microcontroller |
| Pico W | RP2040 Dual M0+ | 133 MHz | 264KB | 2MB Flash | \$6 | Tiny MCU + WiFi | IoT MCU |

Raspberry Pi 5

Features

- Processor: Broadcom BCM2712
- CPU: Quad-core ARM Cortex-A76 @ 2.4GHz
- GPU: VideoCore VII @ 800MHz
- RAM: LPDDR4X (2GB/4GB/8GB/16GB options)
- Storage: MicroSD + M.2 NVMe support (via HAT)
- USB: 2x USB 3.0, 2x USB 2.0
- Display: Dual 4K@60fps HDMI
- Networking: Gigabit Ethernet, WiFi 5, Bluetooth 5.0
- GPIO: 40-pin header
- Power: 5V/5A USB-C (27W recommended)
- Performance: 2-3x faster than Pi 4
- Price: \$60 (2GB), \$80 (4GB), \$100 (8GB), \$150 (16GB)



Note

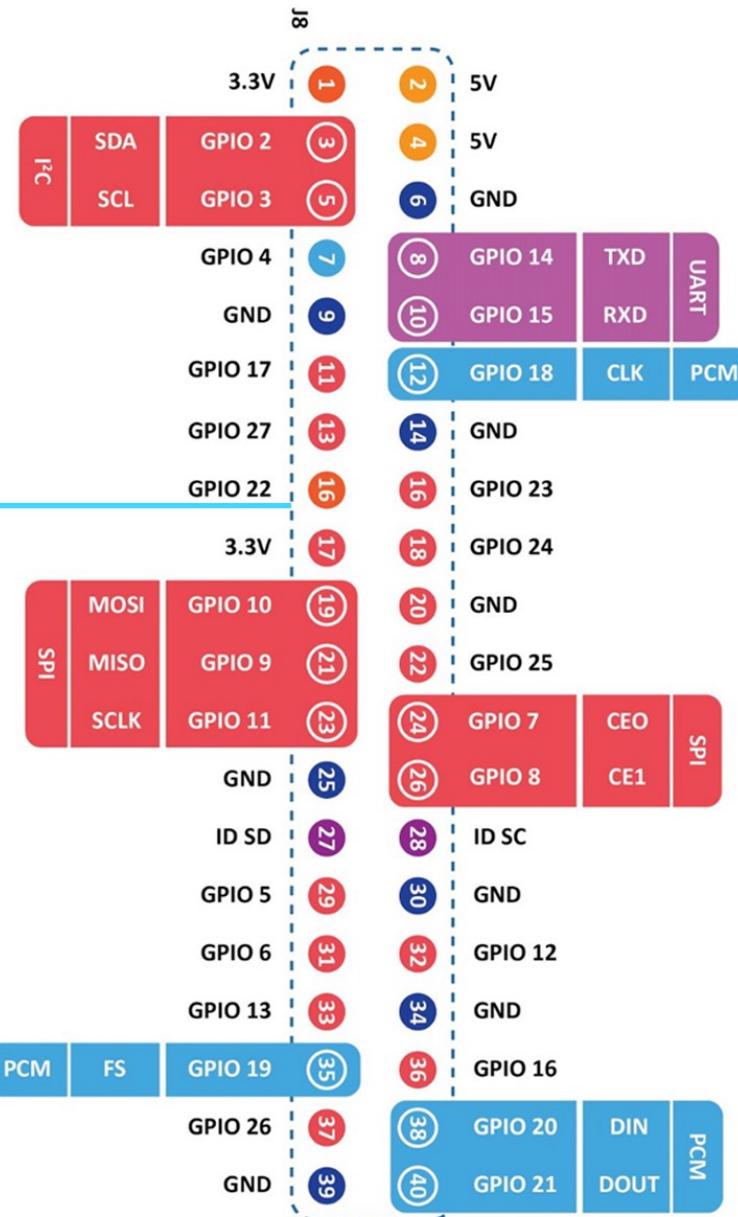
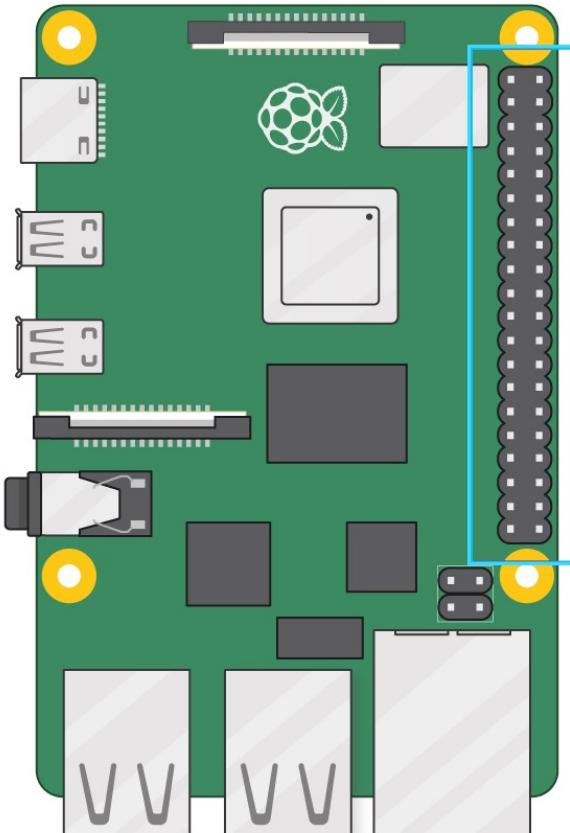
All dimensions in mm

All dimensions are approximate and for reference purposes only.
The dimensions shown should not be used for producing
production data

The dimensions are subject to part and manufacturing tolerances

Not all of the board components are shown. Please refer to the physical board for representation of componentry

Dimensions may be subject to change



Kesimpulan

- Untuk keperluan pembelajaran pembuatan sistem terbenam, Arduino menawarkan ekosistem yang lebih baik dan penggunaannya yang sederhana
- Untuk keperluan IoT yang lebih murah dibandingkan Arduino, bisa menggunakan Wemos
- Untuk keperluan komputasi yang lebih kompleks layaknya penggunaan komputer, Raspberry Pi bisa menjadi pilihan terbaik



DIKTISAINTEK
BERDAMPAK

Pertemuan 6

Pengantar Robotika

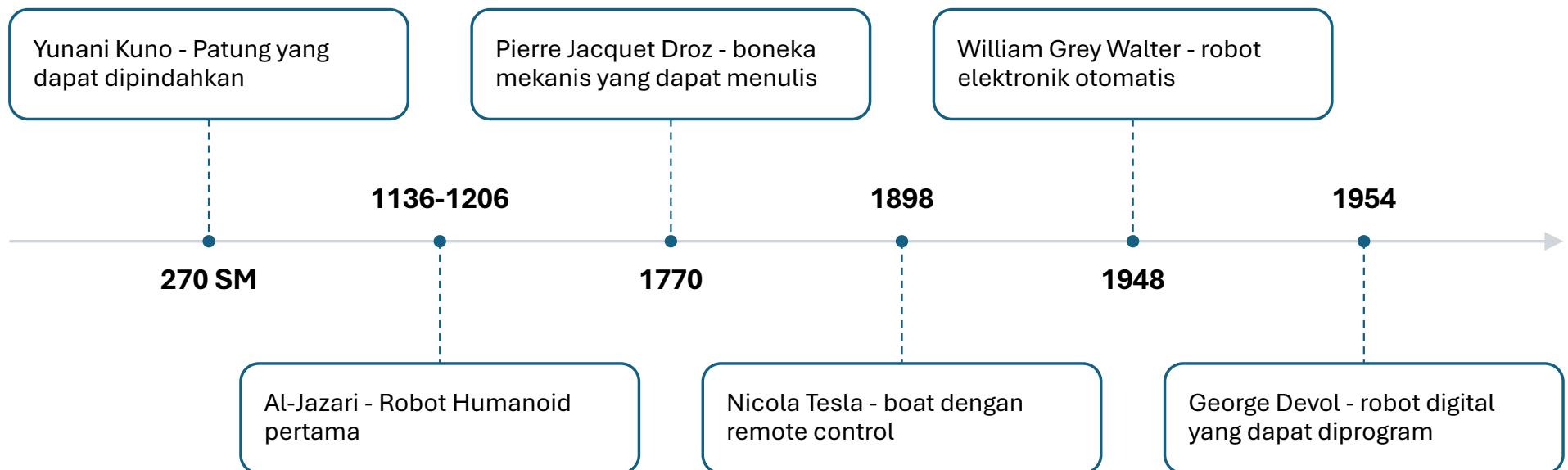
Ali Idrus, S.Kom., M.Kom.

Muhammad Ridho Kurniawan Pratama, S.Kom., M.T.I.

Tujuan pembelajaran

- Memahami sejarah robotika
- Menjelaskan karakteristik dan kinematika robot
- Mengenal komponen dasar robotika
- Memahami prinsip kerja sensor dan aktuator

Sejarah Robotika



Karakteristik Robot

Robot adalah **rancangan yang membantu pekerjaan manusia** dengan karakteristik sebagai berikut

- **Sensing** - Mendeteksi lingkungan (halangan, panas, suara, gambar)
- **Mampu Bergerak** - Menggunakan kaki, roda, terbang, atau berenang
- **Cerdas** - Memiliki kecerdasan buatan untuk keputusan tepat
- **Power** - Membutuhkan daya memadai untuk pengontrol dan aktuator

Klasifikasi Robot

1. Robot Mobile

- Menggunakan roda untuk bergerak
- Contoh: Line Follower, Maze Solving

2. Robot Jaringan

- Dikontrol melalui internet (TCP/IP)
- Contoh: LIPI Wireless Robot (LWR)

3. Robot Manipulator

- Memiliki tangan seperti manusia
- Contoh: Robot las industri, robot perakit

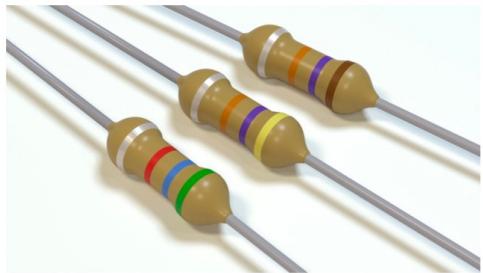
4. Robot Humanoid

- Menyerupai tubuh manusia
- Dilengkapi sensor, aktuator, dan sistem kontrol

5. Robot Terbang

- Bergerak di udara
- Dapat dikontrol wireless atau otomatis
- Contoh: drone

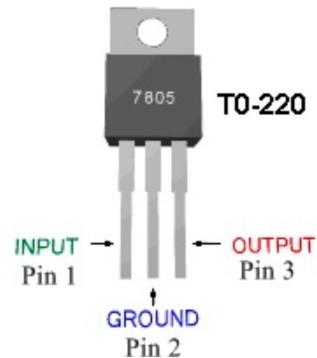
Komponen dasar robot



Resistor - menghambat arus listrik



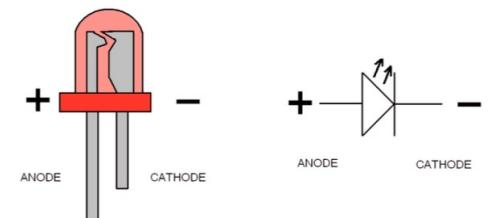
Kapasitor – menyimpan muatan listrik



IC Regulator - menstabilkan tegangan (contoh: LM7805)

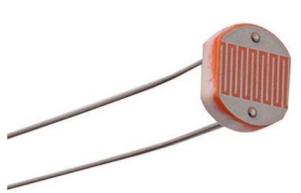


Dioda – mengalirkan arus satu arah



LED – indikator cahaya

Beberapa jenis sensor



Sensor Cahaya – LDR & Photodiode



PIR – mendeteksi keberadaan manusia



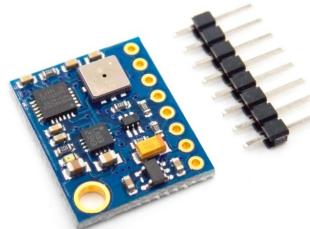
Ultrasonic – mengukur jarak ke objek lain



UVTRON – Mendeteksi nyala api



Kamera – Menangkap citra gambar



Kompas (HMC5883) – Menentukan arah



Gyro – Mendeteksi orientasi gerak

Aktuator: Motor DC

Prinsip Kerja

- Mengubah energi listrik menjadi energi mekanik
- Menggunakan prinsip gaya Lorentz
- Terdiri dari rotor (berputar) dan stator (tetap)



Jenis Motor DC

- Motor DC Magnet Tetap
- Motor Brushless (BLDC)
- Motor Servo
- Motor Stepper



Aktuator: Motor Servo

Karakteristik

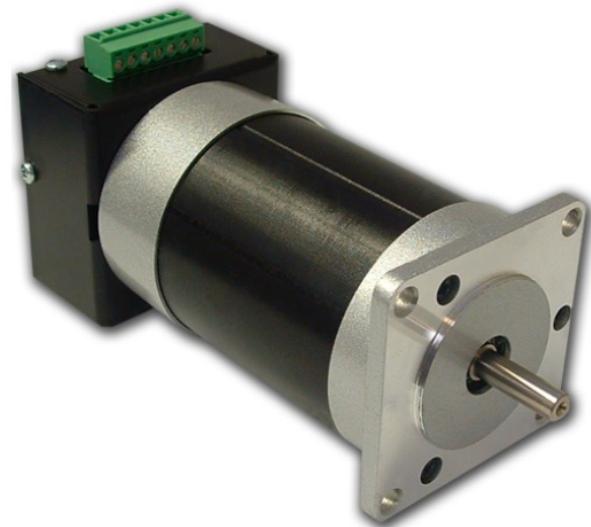
- Kontrol posisi presisi menggunakan PWM
- Terdiri 3 kabel: Vcc, Gnd, Kontrol
- Jenis: Continuous (360°) dan Standard (180°)



Aktuator: Motor Stepper

Karakteristik

- Bergerak per step dengan berbagai resolusi (besar derajat per step)
- Torsi besar pada kecepatan rendah



Aktuator: Solenoid

Karakteristik

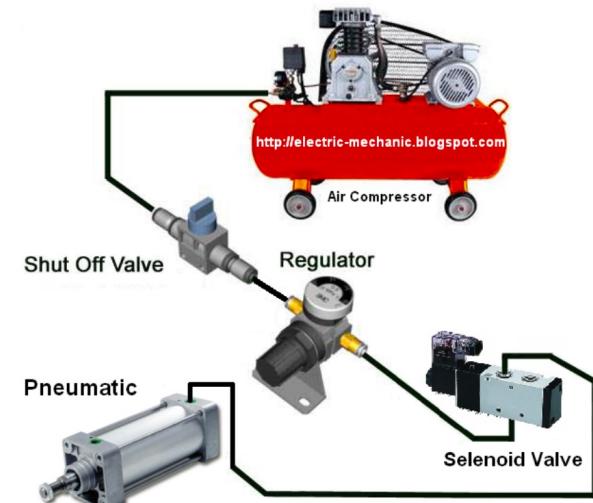
- Gerakan maju-mundur
- Menggunakan medan magnet
- Aplikasi: valve pneumatic, doorlock



Aktuator: Pneumatic

Karakteristik

- Menggunakan tekanan udara
- Menggerakkan cylinder kerja untuk mengubah tekanan udara menjadi tenaga mekanik



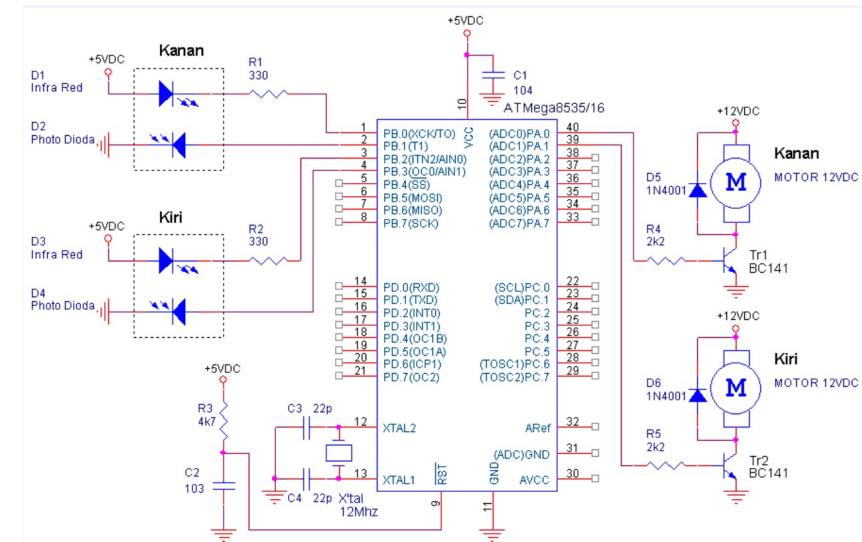
Kegunaan mikrokontroler pada robot

Fungsi

- Otak pengendali robot
- Memproses input dari sensor
- Mengendalikan aktuator sesuai program

Contoh Aplikasi - Line Follower

- Menggunakan sensor photodiode
- Mengikuti garis hitam di lantai putih
- Algoritma dasar: Sensor → Mikrokontroler → Driver Motor → Aktuator



Contoh aplikasi: robot pemadam api

Mikrokontroler: Arduino Mega / STM32

Sensor:

- Sensor Api (UVTRON / Flame Sensor)
- Sensor Ultrasonic (pendeksi dinding)
- Sensor Suhu (LM35)

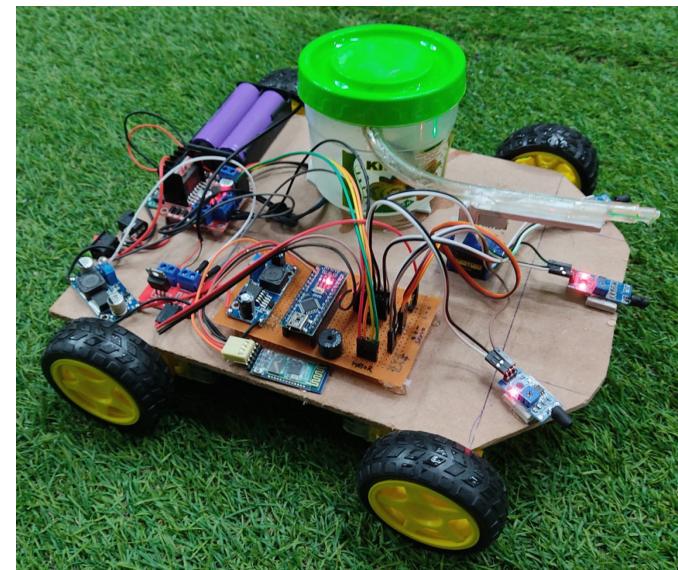
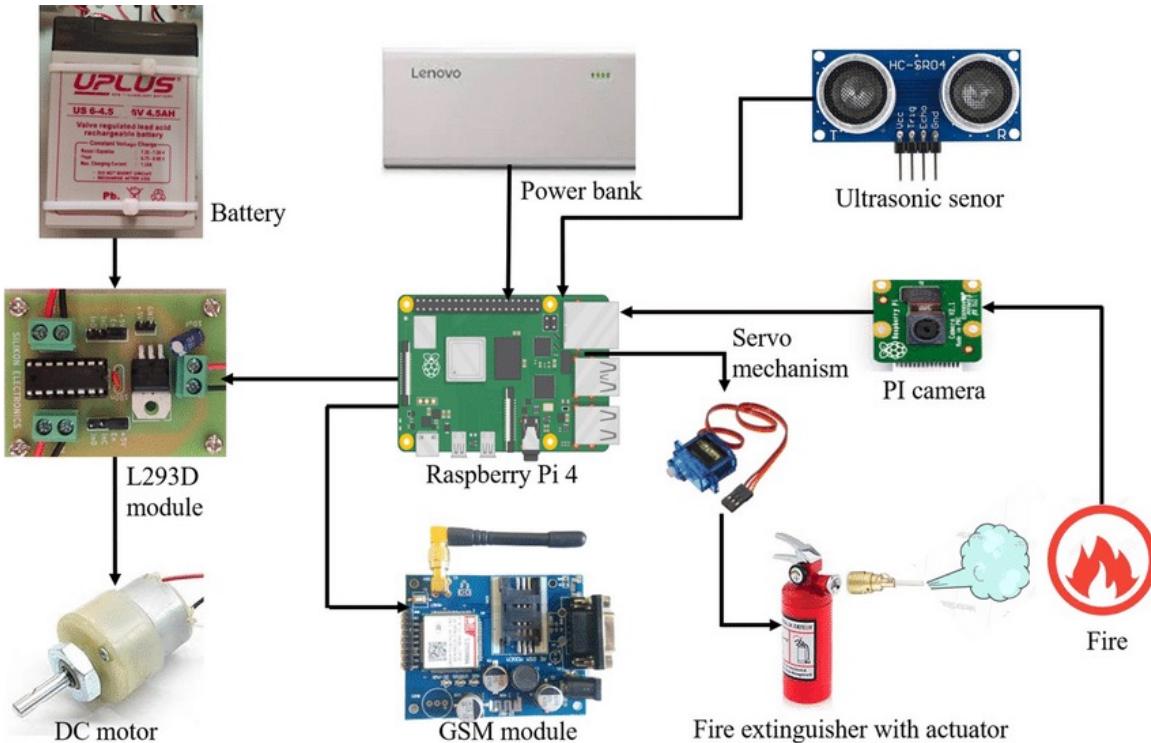
Driver: L293D atau relay module

Power: Battery 12V

Aktuator:

- Motor DC untuk mobilitas
- Mini water pump / kipas DC
- Motor servo untuk arah penyemprot

Karakteristik: navigasi otonom mencari sumber api & memadamkannya



Gambar:

https://www.researchgate.net/figure/Operational-block-diagram-of-firefighting-robot-The-fire-is-detected-by-Raspberry-Pi_fig1_355033608
<https://techpacs.ca/diy-fire-fighting-robot-using-arduino-and-android-app-control-2214>

Kinematika robot

Definisi: Studi pergerakan robot tanpa memperhatikan gaya

Jenis Kinematika

- Forward Kinematics - Dari panjang link & sudut joint → posisi robot
- Inverse Kinematics - Dari panjang link & posisi → sudut joint

Kategori

- Manipulator Kinematics
- Parallel Kinematics
- Mobile Robot Kinematics
- Humanoid Kinematics

Kesimpulan

Komponen Utama Robot

- Power Supply - Sumber daya
- Sensor - Indra robot (sensing)
- Mikrokontroler - Otak pengendali
- Aktuator - Penggerak

Prinsip Kerja: Input (Sensor) →

Proses (Mikrokontroler) → Output
(Aktuator)

Robot dirancang untuk membantu
pekerjaan manusia dengan efisien
dan akurat.



DIKTISAINTEK
BERDAMPAK

Pertemuan 7

Aplikasi Sistem Tertanam Pada Robotika (lanjutan)

Ali Idrus, S.Kom., M.Kom.

Muhammad Ridho Kurniawan Pratama, S.Kom., M.T.I.

Tujuan pembelajaran

- Memahami embedded sistem path planning
- Mengenal sistem minimum (sismin)
- Memahami fitur path planning
- Mengenal akses kontrol pintu dengan sidik jari
- Mengidentifikasi komponen utama sistem

Robot Path Planning + Gripper

Definisi: Sistem robot yang diatur dalam bentuk path untuk memindahkan barang secara otomatis

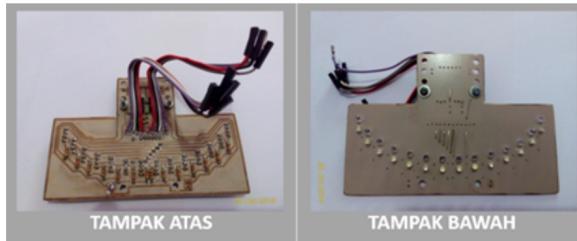
Arsitektur Sistem:

- INPUT → PROSES → OUTPUT
- Sensor → ATmega32 → Aktuator

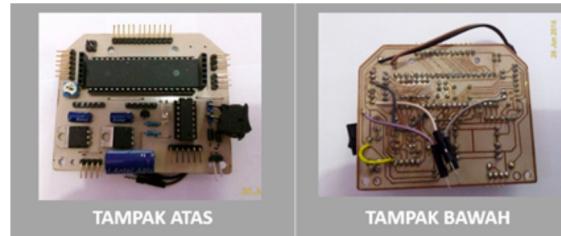
Komponen Utama:

- **Input:** Sensor Proximity (14 buah)
+ Push Button
- **Proses:** Mikrokontroler ATmega32
- **Output:** Motor DC + Motor Servo
(Gripper) + LCD 16x2

Komponen Robot Path Planning



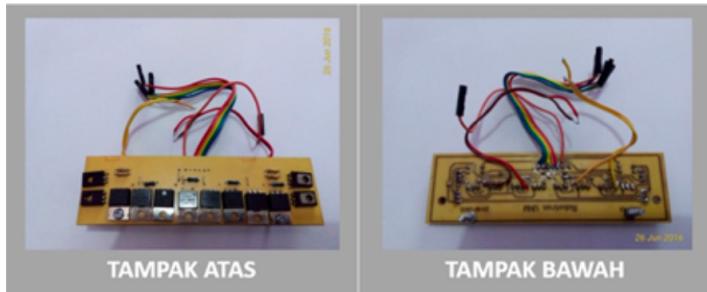
Gambar 7.2: Rangkaian Sensor



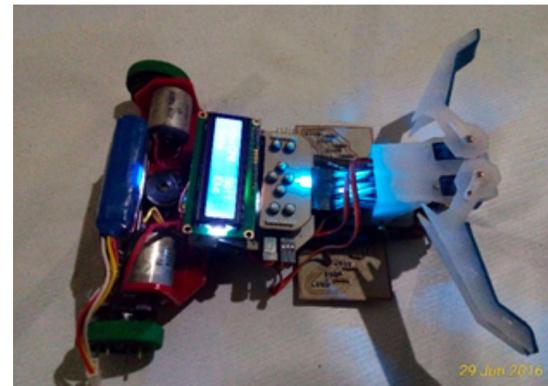
Gambar 7.3: Rangkaian SisMin



Gambar 7.4: Rangkaian Tombol



Gambar 7.5: Rangkaian Driver



Gambar 7.6: Desain Robot

Sistem minimum

Definisi: Pusat rangkaian pengontrol robot berisi IC mikrokontroler

Fungsi Utama:

- Menerima data analog dari sensor (Port A - ADC)
- Memproses input tombol (Port C)
- Menampilkan ke LCD (Port B)
- Mengirim sinyal PWM ke driver motor

Komponen:

- Aktif: Dioda, Transistor BD139
- Pasif: Resistor, Trimmer, Kapasitor, Elko

Rangkaian Sensor & Driver

Sensor Proximity:

- Photodiode 3mm (receiver)
- LED Super Bright (transmitter)
- Membaca jalur hitam di lantai putih
- Dibagi: Sensor Kiri & Kanan

Driver Motor:

- Komponen: IRF9640, IRFZ44N, BD139
- Fungsi: Mensuplay arus besar
- Mengontrol arah & kecepatan motor

Fitur Path Planning

12 Arah Pergerakan:

- Stop, Lurus, Kanan, Kiri
- Am-Ka/Ki (Ambil barang kanan/kiri)
- Am-LKa/LKi (Ambil lurus lalu belok)
- Sm-Ka/Ki (Simpan barang kanan/kiri)
- Sm-LKa/LKi (Simpan lurus lalu belok)

Parameter Path (0-99): Sensor | Kecepatan | Belokan | ErrorPID | Timer | Break

Fitur Tambahan: Auto kalibrasi, Insert/Delete path, Looping, Reset factory

Menu & Navigasi

Menu Utama:

- GO!CP1: Jalan dengan path terprogram
- GO: Free run tanpa path
- Sens: Cek sensor (ADC + bar sensor)
- Etc: Factory reset, delay config

Pengaturan Free Run: Kecepatan,

Sensor, Turbo Break, Kecepatan maju/mundur

Kalibrasi Sensor: Atur ADC manual atau auto kalibrasi

Akses Kontrol Pintu

Sistem Keamanan Biometrik dengan Sidik Jari

Hardware Utama:

- Arduino Uno (brain)
- Sensor Ultrasonik HC-SR04 (trigger)
- Sensor Fingerprint FZ1035 (verifikasi)
- Solenoid Door Lock (pengunci)
- Motor Servo (pembuka pintu)
- LCD 16x2 + Push Button + Saklar Limit

Dimensi Pintu: 1m x 1.5m | Max 15 Kg

Sensor Sistem

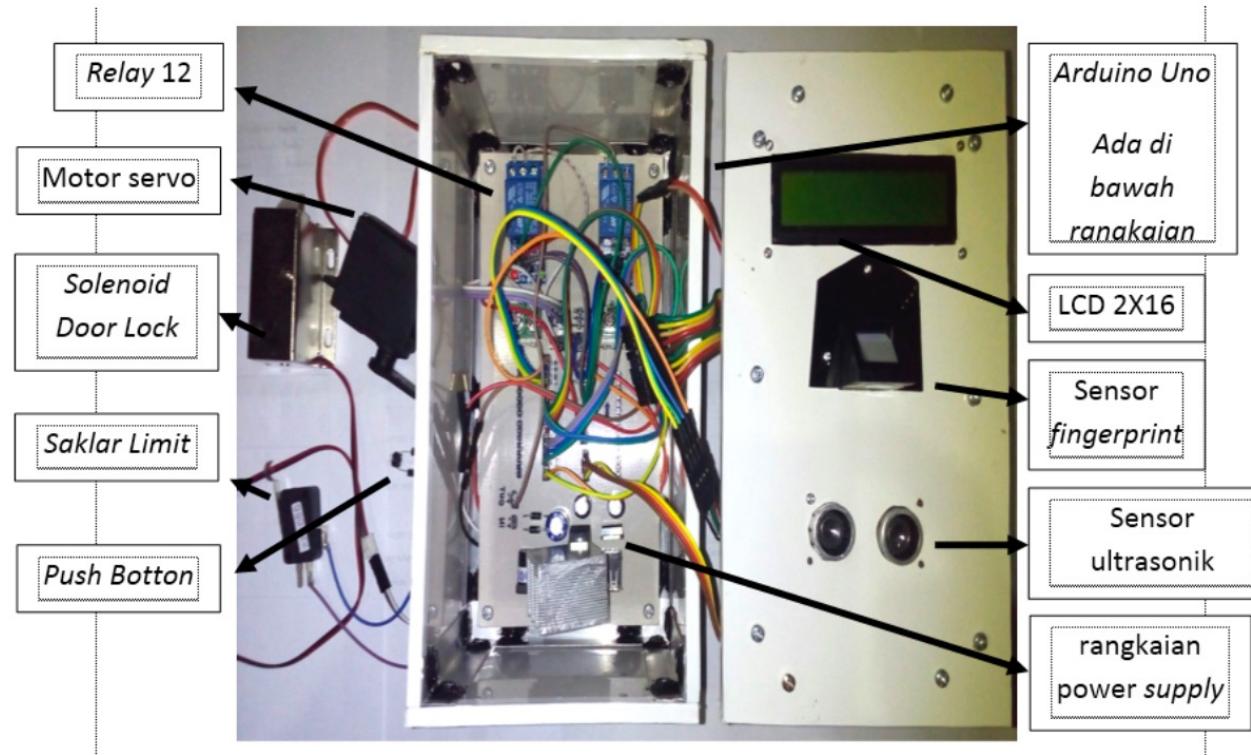
Sensor Ultrasonik HC-SR04

- Tegangan: 5 VDC
- Jarak aktif: ≤ 30 cm
- Fungsi: Trigger aktivasi fingerprint

Sensor Fingerprint FZ1035

- Tegangan: 5 VDC
- Pin: Vcc, GND, RxD, TxD
- Fungsi: Scan & verifikasi sidik jari
- Indikator: Berkedip saat siap scan

Akses Kontrol Pintu



Gambar 7.15: Akses Kontrol Pintu Menggunakan Sidik Jari

Modul Relay & Aktuator

2 Relay 12 VDC

Relay 1 → Solenoid Door Lock

- Input: 12 VDC
- NO (Normally Open): Pintu terbuka
- Aktif: Pengunci terpental masuk

Relay 2 → Motor Servo:

- Input: 5 VDC
- Sudut: 90° (pintu terbuka)
- Torsi: Maksimal 15 Kg

Komponen Relay: Resistor, Transistor, Dioda (proteksi)

Kontrol & Reset

Push Button (Dalam Ruangan):

- Membuka pintu dari dalam
- Bypass sistem fingerprint
- Emergency access

Saklar Limit:

- Posisi: Pintu terbuka 90°
- Fungsi: Reset sistem
- Efek: Putus tegangan servo
- Hasil: Pintu mudah ditutup manual

Display

LCD 16x2 Display

Tampilan

- "System | Pintu Otomatis" (awal)
- "Silahkan Scan" (standby)
- "Silahkan Masuk" (akses OK)
- "Akses Ditolak" (fingerprint tidak cocok)
- "Pintu Terbuka" (status)

Koneksi: 6 pin ke Arduino Uno



Gambar 7.22: Display Keadaan Pintu Terbuka

Konfigurasi Pin

11 Pin Digital I/O:

INPUT:

- Reset: Saklar limit
- A1: Push button
- A4/A5: Echo/Trigger ultrasonik
- 6/7: TxD/RxD fingerprint

OUTPUT:

- 2,3,4,5: Data LCD
- 11,12: Reset/Enable LCD
- A2: Kontrol solenoid
- A3: Kontrol motor servo

Power Supply

Dual Adaptor (Terpisah):

5 VDC:

- Arduino Uno
- Motor Servo
- Sensor Fingerprint
- LCD 16x2

12 VDC:

- Relay (2 unit)
- Solenoid Door Lock

Alasan: Mencegah overheating IC regulator untuk operasi 24 jam