

MATERI SISTEM KONTROL DAN AUTOMASI
PEMBELAJARAN TEKNOLOGI INFORMASI DAN ELEKTRONIKA



Oleh

Nandito Fernandes Hornay,.S.T

DEPARTAMENTO ELETRONIKA
ESCOLA TÉCNICA VOCACIONAL – GRUPO DE TECNOLOGIA E INDÚSTRIA

2025

MATERI SISTEM KONTROL DAN AUTOMASI

Tujuan:

1. Memahami definisi, ruang lingkup, dan manfaat sistem kontrol dan automasi.
2. Mengetahui dan merancang komponen dasar sistem kontrol dan automasi.

Definisi Kontrol dan Automasi

Kontrol adalah proses pengaturan perilaku suatu sistem agar mencapai kondisi atau keluaran yang diinginkan. Dalam konteks elektronika dan otomasi, kontrol umumnya melibatkan pengukuran sinyal masukan (misalnya suhu, posisi, atau kecepatan) dan pemberian sinyal keluaran (misalnya tegangan, arus, atau pulsa PWM) melalui mikrokontroler ATmega8 atau arduino Uno untuk memengaruhi perangkat. Dengan kontrol, sistem dapat mempertahankan kestabilan meski terjadi gangguan atau perubahan beban. Automasi adalah penerapan teknik kontrol secara otomatis tanpa intervensi manusia berulang-ulang. Automasi memanfaatkan mikrokontroler ATmega8 atau arduino uno sebagai “otak” yang mengeksekusi algoritma kontrol mulai dari logika sederhana on/off hingga pengendalian PID (*Proportional-Integral-Derivative*) berdasarkan data sensor. Hasilnya, proses industri atau laboratorium dapat berlangsung lebih konsisten, efisien, dan aman. Pada sistem kontrol terbuka (open-loop), mikrokontroler mengirimkan sinyal keluaran berdasarkan program tetap tanpa umpan balik (feedback) dari sensor.

Contoh sederhananya adalah timer otomatis lampu, di mana lampu menyala selama interval yang diatur tanpa memeriksa kondisi lingkungan. Sementara itu, sistem tertutup (closed-loop) memanfaatkan sensor untuk mengukur hasil keluaran, lalu menyesuaikan perintah melalui algoritma kontrol, misalnya menjaga suhu di 50 °C dengan memodulasi daya pemanas. Penggabungan kontrol dan automasi menghasilkan sistem kendali otomatis yang mampu merespon perubahan kondisi lingkungan secara adaptif. Misalnya, mikrokontroler dapat membaca pembacaan LDR (sensor cahaya) dan mengatur intensitas lampu LED secara otomatis. Teknik ini meminimalkan kesalahan manusia dan meningkatkan ketelitian pengujian. Dengan memahami definisi kontrol dan automasi, siswa elektronik STM akan mampu merancang rangkaian yang tepat mulai pemilihan sensor, pemrograman register ADC/DIO pada mikrokontroler, hingga pengaturan sinyal PWM sehingga sistem kendali menjadi efektif, efisien, serta sesuai kebutuhan aplikasi industri atau eksperimen laboratorium.

Sejarah dan Perkembangan Otomasi

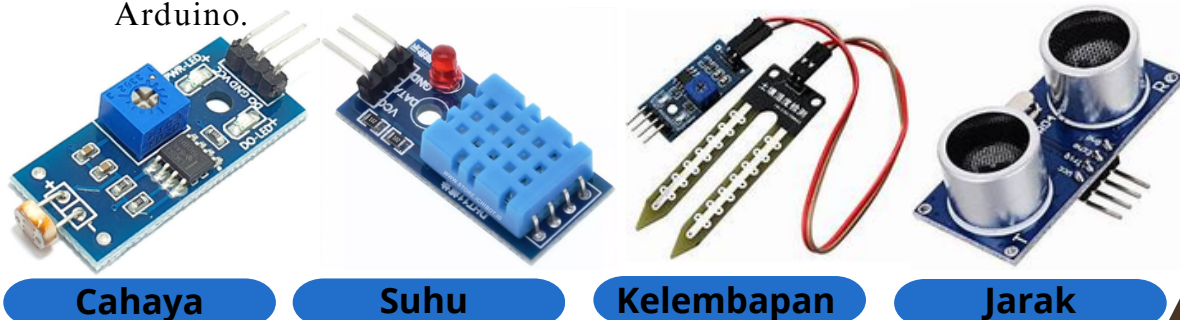
Awal mula dari otomasi industri dimulai pada Revolusi Industri ke-1 (abad ke-18), ketika tenaga uap menggantikan tenaga manusia dan hewan. Kendali masih manual; namun, lahirnya peralatan mekanis membuka jalan bagi gagasan pengaturan otomatis. Perkembangan transistor dan sirkuit terpadu di abad ke-20 kemudian memungkinkan desain sistem kontrol elektronik pertama. Pada Revolusi Industri ke-2, listrik dan assembly line Ford pabrik mobil memperkenalkan konsep produksi massal dengan mekanisme conveyor. Kendali pneumatik dan hidrolik menggantikan proses manual, tetapi fleksibilitas masih terbatas. Sensor analog sederhana mulai digunakan, meski tanpa mikrokontroler canggih seperti mikrokontroler (ATmega8 atau Arduino).

Revolusi Industri ke-3 (dikenal sebagai era otomasi digital) muncul pada 1970-an dengan ditemukannya PLC (Programmable Logic Controller). PLC menjadi cikal bakal mikrokontroler dalam industri, mengotomasi logika on/off dan sekuensial. Di kalangan pendidikan STM, ATmega8 hadir sebagai “PLC mini” untuk eksperimen, memudahkan siswa memahami konsep dasar pemrograman dan I/O digital. Seiring era Industri 4.0, konektivitas IoT, AI, dan big data mengubah lanskap otomasi. Mikrokontroler sederhana seperti ATmega8 tetap relevan untuk prototipe dan pendidikan dasar, sebelum beralih ke platform lebih canggih (STM32, ESP32). Namun, prinsip kendali dan automasi yang dipelajari pada ATmega8 masih menjadi fondasi pemahaman sistem kompleks. Dengan menelusuri sejarah otomasi, siswa dapat menghargai evolusi teknologi from rel mekanik ke kontrol digital serta memahami mengapa mikrokontroler seperti ATmega8 sangat penting dalam pengembangan sistem kendali modern, terutama di lingkungan laboratorium dan bengkel elektronik STM.

Elemen Dasar

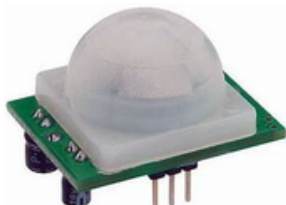
1

Sensor adalah sebuah komponen atau perangkat yang digunakan untuk mendeteksi perubahan fisik atau lingkungan seperti cahaya, suhu, kelembapan, tekanan, jarak, gerakan, suara, dan sebagainya, kemudian mengubahnya menjadi sinyal listrik yang dapat dibaca dan diproses oleh sistem elektronik, terutama oleh mikrokontroler seperti Arduino.





Sensor Api



Sensor Gerak



Sensor Suara



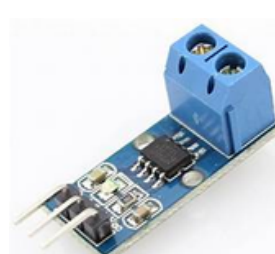
Sensor Warna



Sensor Hujan



Sensor Getar



Sensor Arus

2

LCD dan I2C

LCD (Liquid Crystal Display) yaitu digunakan untuk menampilkan data atau informasi dari mikrokontroler seperti Arduino Uno.

Namun, sering kali dalam pembelajaran atau istilah yang salah kaprah, orang menyebut “sensor LCD” padahal yang dimaksud adalah “LCD untuk menampilkan data dari sensor”.

I2C (dibaca: "I-squared-C" atau "I-two-C") adalah singkatan dari Inter-Integrated Circuit, yaitu sebuah protokol komunikasi serial yang digunakan untuk pertukaran data antara mikrokontroler (seperti Arduino atau ATmega) dengan perangkat lain seperti sensor, LCD, EEPROM, dan modul lainnya hanya melalui dua jalur data:

- SDA (Serial Data Line) – jalur data
- SCL (Serial Clock Line) – jalur clock



LCD



I2C



LCD dan I2C

3

Komponen elektronika adalah bagian-bagian dasar yang digunakan dalam rangkaian elektronika untuk mengontrol aliran arus listrik, memproses sinyal, menyimpan energi, atau melakukan fungsi-fungsi tertentu seperti memperkuat sinyal, menghasilkan frekuensi, dan lain-lain.

Jenis-Jenis Komponen Elektronika

1. Komponen Aktif
2. Komponen Pasif
3. Komponen Elektromekanik

1. Komponen Aktif

Komponen yang membutuhkan sumber daya eksternal untuk beroperasi dan dapat menguatkan sinyal.

Contoh:

- Dioda: Mengalirkan arus hanya dalam satu arah.
- Transistor: Sebagai saklar atau penguat.
- IC (Integrated Circuit): Gabungan banyak komponen dalam satu chip.

2. Komponen Pasif

Komponen yang tidak dapat memperkuat sinyal dan tidak memerlukan sumber daya eksternal.

Contoh:

- Resistor: Menghambat aliran arus listrik.
- Kapasitor: Menyimpan dan melepaskan muatan listrik.
- Induktor (Koil): Menyimpan energi dalam bentuk medan magnet.

3. Komponen Elektromekanik

Komponen yang menggabungkan elemen listrik dan mekanik.

Contoh:

- Saklar (Switch): Menghubungkan atau memutuskan arus listrik secara manual.
- Relay: Saklar elektromagnetik.
- Potensiometer: Resistor yang bisa diatur nilainya.

Gambar Komponen Pasif



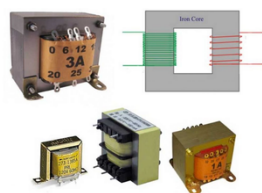
Resistor



Kapasitor



Induktor



Transformator

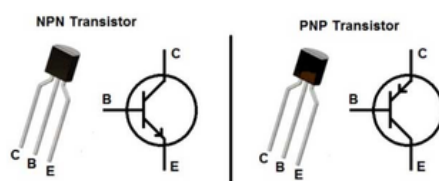


Thermistor

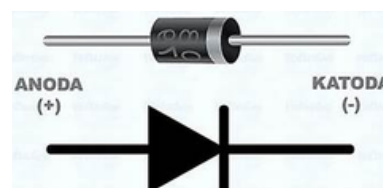


LDR

Gambar Komponen Aktif



Transistor



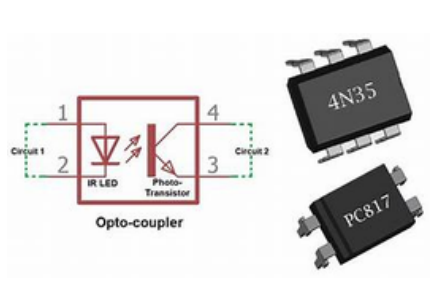
Dioda



IC (integrated Circuit)



Thyristor (TRIAC)



Optocoupler

 **Gambar Komponen Komponen Elektromekanik**



Relay



Solenoid



Saklar



Motor DC

4

Pengendali

Pengendali mikrokontroler ATmega adalah unit inti dalam sistem berbasis mikrokontroler yang berfungsi sebagai otak utama untuk mengatur, memproses, dan mengendalikan seluruh operasi perangkat elektronik sesuai dengan program yang telah ditanamkan. ATmega, seperti ATmega8 atau ATmega328, merupakan keluarga mikrokontroler dari AVR yang memiliki arsitektur RISC 8-bit dan dilengkapi dengan fitur seperti memori flash, EEPROM, RAM, serta antarmuka input/output digital dan analog, sehingga memungkinkan pengendalian berbagai perangkat seperti sensor, aktuator, motor, dan tampilan display secara efisien dan terprogram.

 **Gambar Pengendali Mikrokontroler**



Atmel AVR



AVR



ATX Mega



ATmega 328P



PIC 18F877A



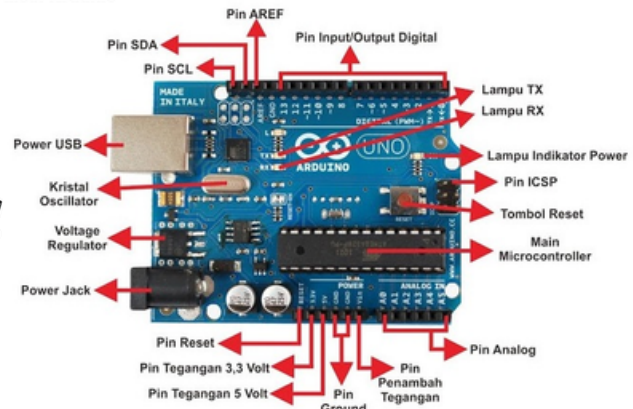
8051



Arduino



ARM



Arduino Uno adalah salah satu papan mikrokontroler yang paling populer dan banyak digunakan dalam pembelajaran dan proyek elektronika.

Arduino Uno R3



Karakteristik Pada Arduino Uno



Mikrokontroler Utama

- Tipe: ATmega328P
- Frekuensi Clock: 16 MHz
- Mikrokontroler ini adalah "otak" dari papan Arduino Uno yang menjalankan program (sketch) yang diunggah dari komputer.



Memori

- Flash Memory: 32 KB (termasuk 0,5 KB digunakan oleh bootloader)
- SRAM: 2 KB
- EEPROM: 1 KB
- Memori ini digunakan untuk menyimpan kode program dan variabel saat Arduino berjalan.



Input/Output (I/O)

- Digital I/O: 14 pin (6 di antaranya bisa digunakan sebagai PWM)
- Analog Input: 6 pin (A0–A5)
- Pin PWM: 3, 5, 6, 9, 10, 11
- Pin Komunikasi Serial: TX (1), RX (0)
- Tersedia juga pin I2C (A4 - SDA dan A5 - SCL) serta SPI (pin 10–13).



Koneksi dan Catu Daya

- Tegangan Operasional: 5V
- Tegangan Input (disarankan): 7–12V
- Konektor USB Type B untuk pemrograman dan komunikasi dengan komputer
- Jack DC untuk catu daya eksternal



Fitur Tambahan

- Komunikasi Serial: UART, I2C, dan SPI
- Port USB: Tipe B, digunakan untuk pemrograman dan komunikasi serial
- Regulator Tegangan: Input bisa dari USB (5V) atau jack DC (7–12V)
- Konektor Power:
 - Vin (tegangan input eksternal)
 - 5V (output tegangan stabil 5V)
 - 3.3V (output 3.3V)
 - GND (ground)
- Crystal Oscillator: 16 MHz



Lingkungan Pengembangan

- Menggunakan *Software* Arduino IDE
- Bahasa pemrograman berbasis C/C++ sederhana

Tipe Sistem Kontrol (Open-Loop, Closed-Loop)

1

Open loop adalah sistem kontrol tanpa umpan balik. Output tidak mempengaruhi input. Sistem ini bekerja berdasarkan perintah awal tanpa menyesuaikan diri terhadap perubahan lingkungan atau hasil proses.

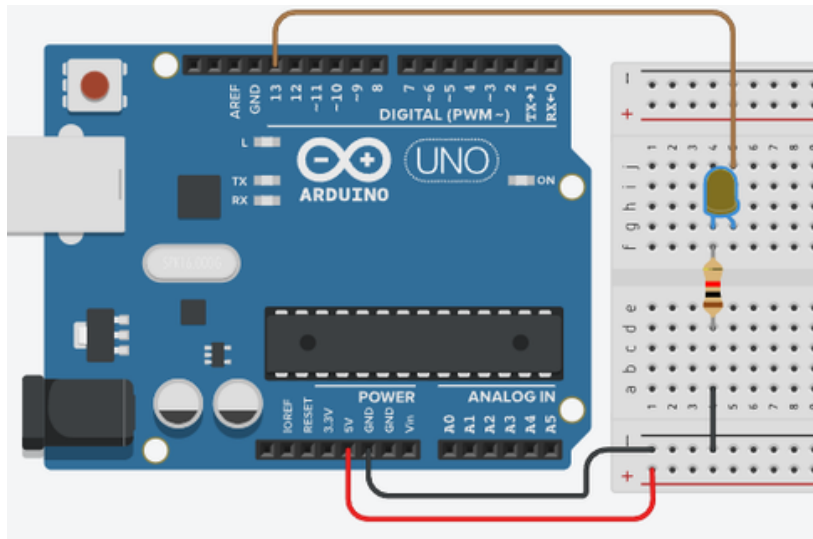
Contoh: Menyalakan LED selama 5 detik, lalu mati. Tidak peduli apakah LED benar-benar menyala atau tidak.



Source Code:

```
// Konfigurasi untuk ATmega8 di Arduino IDE
void setup()
{ pinMode(13, OUTPUT); // LED pada pin 13
}
void loop() { digitalWrite(13, HIGH); // LED menyala delay(5000);
digitalWrite(13, LOW); // LED mati while (1); // Hentikan program
(sekali saja menyala)
}
```


Rangkaian



2

Closed loop menggunakan umpan balik (feedback) untuk mengontrol output. Sistem ini terus menerus memantau hasil dan menyesuaikan input agar output sesuai yang diharapkan.

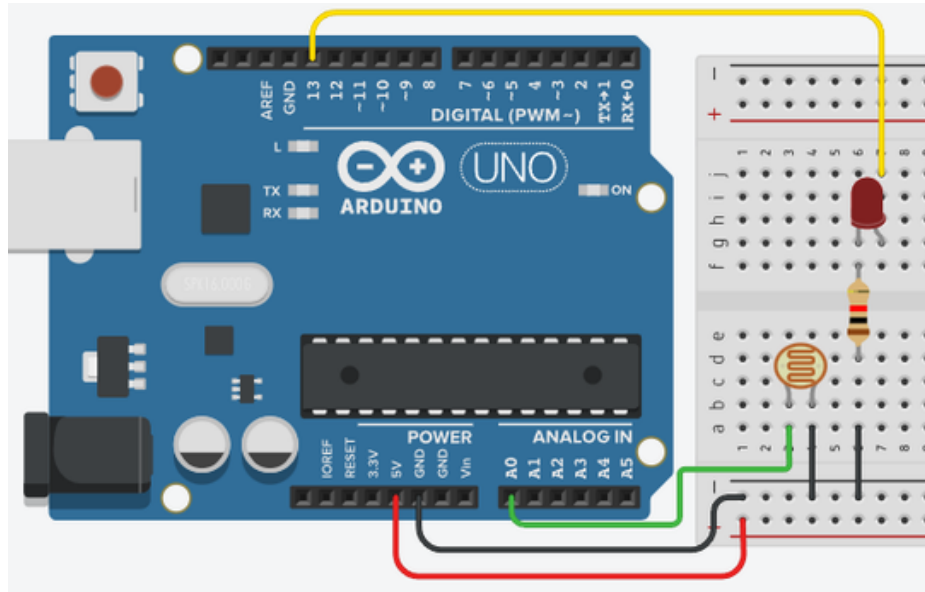
Contoh: LED akan menyala saat lingkungan gelap, dan mati saat terang menggunakan LDR (Light Dependent Resistor) sebagai sensor cahaya.



Source Code:

```
const int ldrPin = A0;    // pin input sensor LDR
const int ledPin = 13;    // pin output untuk LED
int nilaiCahaya;          // variabel untuk menyimpan pembacaan LDR
void setup() {
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);      // untuk melihat data di Serial Monitor
                           // (opsional)
}
void loop() {
  nilaiCahaya = analogRead(ldrPin); // baca nilai dari LDR (0 - 1023)
  Serial.println(nilaiCahaya);      // tampilkan nilai di Serial Monitor
  if (nilaiCahaya < 500) { // jika gelap (nilai kecil)
    digitalWrite(ledPin, HIGH); // nyalakan LED
  } else {
    digitalWrite(ledPin, LOW); // matikan LED
  }
  delay(100); // delay sedikit untuk stabilisasi pembacaan
}
```

Rangkaian



Contoh Penerapan

 **Pemrograman Arduino UNO menggunakan Arduino IDE untuk menyalakan LED berdasarkan jarak menggunakan sensor Ultrasonik**

Contoh kode program Arduino (Arduino Uno IDE) dan penjelasan cara kerja sistem sensor ultrasonik mengontrol LED. Sistem ini akan menyalakan LED ketika jarak objek lebih dekat dari ambang batas tertentu (misalnya 15 cm), dan mematikan LED saat objek menjauh.

Komponen yang Dibutuhkan:

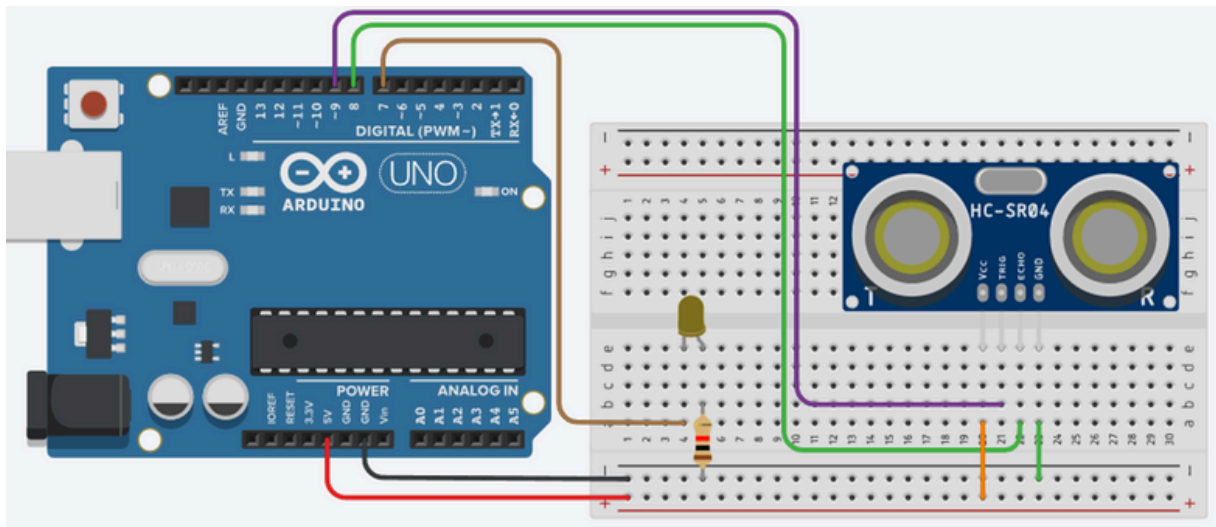
- 1 × Arduino Uno
- 1 × Sensor Ultrasonik HC-SR04
- 1 × LED
- 1 × Resistor 220 ohm
- Kabel jumper
- Breadboard

Skema Rangkaian:

- HC-SR04
 - VCC → 5V Arduino
 - GND → GND Arduino
 - Trig → Pin 9
 - Echo → Pin 10
- LED
 - Anoda (+) → Pin 13
 - Katoda (-) → Resistor 220 ohm → GND



Rangkaian Sensor Ultrasonik



Kode Program Arduino:

```
// Definisi pin
const int trigPin = 9;
const int echoPin = 10;
const int ledPin = 13;

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  long durasi;
  float jarak;
  // Kirim pulsa trigger 10µs
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Baca durasi pantulan echo
  durasi = pulseIn(echoPin, HIGH);

  // Hitung jarak dalam cm
  jarak = durasi * 0.034 / 2;

  Serial.print("Jarak: ");
  Serial.print(jarak);
  Serial.println(" cm");

  // Kontrol LED
  if (jarak < 15) {
    digitalWrite(ledPin, HIGH); // LED nyala jika objek dekat
  } else {
    digitalWrite(ledPin, LOW); // LED mati jika objek jauh
  }
  delay(500); // Delay baca ulang
}
```

Cara Kerja Program:

1. Trigger pin pada HC-SR04 dikirimkan pulsa selama 10 mikrodetik.
2. Sensor akan mengirim gelombang ultrasonik, lalu menunggu pantulannya.
3. Saat gelombang pantul diterima, echo pin menghasilkan pulsa yang durasinya sebanding dengan jarak.

4. Arduino membaca durasi pulsa tersebut, menghitung jarak menggunakan rumus:

$$\text{jarak} = \text{durasi} \times 0.034 / 2$$

5. Jika jarak kurang dari 15 cm, LED menyala (HIGH). Jika lebih jauh, LED padam (LOW).

2

Pemrograman Arduino UNO menggunakan Arduino IDE untuk mengendalikan motor servo

Motor servo adalah aktuator yang digunakan untuk mengatur posisi sudut secara presisi. Servo bekerja berdasarkan sinyal PWM (Pulse Width Modulation) yang dikirim dari Arduino. Panjang pulsa menentukan sudut yang akan dicapai oleh servo.

Biasanya:

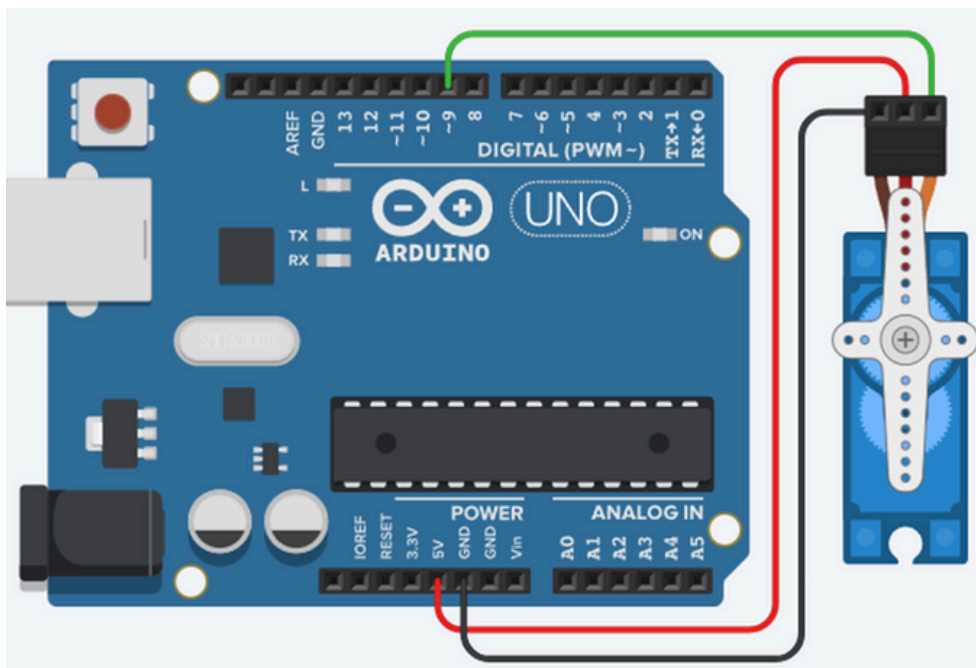
- Pulsa 1 ms = sudut 0°
- Pulsa 1.5 ms = sudut 90°
- Pulsa 2 ms = sudut 180°

Sinyal ini diberikan secara berulang setiap 20 ms. Servo memiliki tiga kabel:

- Merah → VCC (5V)
- Coklat/Hitam → GND
- Kuning/Putih → sinyal dari Arduino (PWM pin)



Rangkaian Motor Servo



Source Code:

Servo Bergerak ke Posisi Tertentu

```
#include <Servo.h>

Servo armServo;

void setup() {
  armServo.attach(9);
  armServo.write(0); //initial starting position
}

void loop() {
  armServo.write(90); //rotate servo to 90 deg
  delay(2000); //pause 2 secs
  armServo.write(0); //rotate servo to 0 deg
  delay(1000); //pause 1 sec
  armServo.write(180); //rotate servo to 180 deg
  delay(2000); //pause 2 secs
}
```

Good
luck