

# Panduan Mengontrol AC Servo Drive P100S Menggunakan Arduino Uno

## Daftar Isi

1. Pengenalan
2. Komponen yang Dibutuhkan
3. Diagram Koneksi
4. Persiapan Servo Drive
5. Konfigurasi Parameter
6. Pemrograman Arduino
7. Pengujian Sistem
8. Troubleshooting
9. Tips dan Saran

## Pengenalan

AC Servo Drive P100S adalah pengendali motor servo yang presisi tinggi. Dokumen ini menjelaskan cara mengontrol servo drive P100S menggunakan Arduino Uno dengan pendekatan yang sederhana dan praktis. Kontrol akan dilakukan melalui mode "Internal Position", yang memungkinkan Anda untuk memprogram hingga 8 posisi berbeda dan memilihnya menggunakan sinyal digital.

## Komponen yang Dibutuhkan

- Arduino Uno
- AC Servo Drive P100S
- Motor Servo yang kompatibel
- Kabel jumper
- Catu daya AC 220V untuk servo drive
- Komponen opsional: resistor pull-up 4.7k $\Omega$  (4 buah)
- Optokopler (jika diperlukan untuk isolasi sinyal)

## Diagram Koneksi

### Koneksi Dasar Arduino ke Servo Drive

Arduino Uno	P100S Servo Drive (CN1 Terminal)
-----	-----
Digital Pin 2	----> DI1 (Pin 20) - CTRG (Trigger)
Digital Pin 3	----> DI2 (Pin 4) - POS0
Digital Pin 4	----> DI3 (Pin 19) - POS1
Digital Pin 5	----> DI4 (Pin 3) - POS2
Digital Pin 6	←---- DO4 (Pin 37) - ZSP (Zero Speed Signal)
GND	----> COM (Pin 16) - Common untuk semua input

## Catatan Penting tentang Koneksi

1. **Isolasi Sinyal:** Disarankan untuk menggunakan optokopler antara Arduino dan Servo Drive untuk melindungi Arduino dari tegangan yang lebih tinggi.
2. **Catu Daya:** Servo Drive memerlukan catu daya AC 220V. JANGAN sambungkan ini ke Arduino!
3. **Level Tegangan:** Input Servo Drive bekerja pada 24V, sementara Arduino mengeluarkan 5V. Perhatikan diagram koneksi lengkap di bawah untuk mengatasi perbedaan ini.

## Koneksi Lengkap dengan Isolasi (Disarankan)

Arduino Uno	Optokopler	Servo Drive (CN1)
-----	-----	-----
Digital 2	--> Input 1	Output 1 --> DI1 (20)
Digital 3	--> Input 2	Output 2 --> DI2 (4)
Digital 4	--> Input 3	Output 3 --> DI3 (19)
Digital 5	--> Input 4	Output 4 --> DI4 (3)
Digital 6	←-- Input 5	Output 5 ←-- DO4 (37)
5V	--> VCC	
GND	--> GND	GND --> COM (16)
		E24V --> Vcc optokopler

## Persiapan Servo Drive

Sebelum menghubungkan ke Arduino, pastikan Servo Drive dikonfigurasi dengan benar:

1. **Hubungkan Servo Motor:** Sambungkan kabel daya motor (U, V, W) dan kabel encoder ke servo drive.
2. **Sambungkan Daya:** Hubungkan catu daya AC 220V ke terminal L1, L2 servo drive.
3. **Pastikan Pengaman:** Pastikan grounding sudah dihubungkan dengan benar untuk keamanan.
4. **Uji Servo:** Uji motor servo dengan mode JOG melalui panel kontrol servo drive:
  - Tekan tombol SET untuk masuk ke menu

- Pilih "JOG" menggunakan tombol ↑ atau ↓
- Tekan SET untuk masuk mode JOG
- Gunakan tombol ↑ atau ↓ untuk menggerakkan motor

## Konfigurasi Parameter

Parameter berikut perlu diubah untuk mengatur servo drive dalam mode internal position. Gunakan panel depan servo drive untuk mengubah parameter:

### Langkah-langkah Mengubah Parameter:

1. Tekan tombol SET untuk masuk ke menu utama
2. Pilih "PA-" (atau kelompok parameter yang sesuai)
3. Tekan SET
4. Pilih nomor parameter menggunakan tombol ↑ atau ↓
5. Tekan SET untuk melihat nilai parameter
6. Ubah nilai menggunakan tombol ↑ atau ↓
7. Tekan SET untuk menyimpan
8. Tekan ← untuk kembali

### Parameter Utama:

1. **Mode Kontrol:**
  - PA4 = 0 (Mode kontrol posisi)
  - PA14 = 3 (Input posisi internal)
2. **Konfigurasi I/O:**
  - P3-0 = 28 (DI1 sebagai CTRG - Trigger perintah posisi)
  - P3-1 = 29 (DI2 sebagai POS0 - Bit 0 pemilihan posisi)
  - P3-2 = 30 (DI3 sebagai POS1 - Bit 1 pemilihan posisi)
  - P3-3 = 31 (DI4 sebagai POS2 - Bit 2 pemilihan posisi)
  - P3-23 = 4 (DO4 sebagai ZSP - Sinyal kecepatan nol)
3. **Konfigurasi Posisi Internal:**
  - P4-0 = 0 (Mode posisi absolut)
  - P4-1 = 0 (Delay output setelah posisi tercapai)
4. **Posisi 1 (POS0=0, POS1=0, POS2=0):**
  - P4-2 = [jumlah putaran] (-30000 hingga 30000)

- P4-3 = [jumlah pulsa dalam putaran]
- P4-4 = [kecepatan dalam r/min] (0-5000)

**5. Posisi 2 (POS0=1, POS1=0, POS2=0):**

- P4-5 = [jumlah putaran]
- P4-6 = [jumlah pulsa dalam putaran]
- P4-7 = [kecepatan dalam r/min]

**6. Posisi 3 (POS0=0, POS1=1, POS2=0):**

- P4-8 = [jumlah putaran]
- P4-9 = [jumlah pulsa dalam putaran]
- P4-10 = [kecepatan dalam r/min]

**7. Posisi 4 (POS0=1, POS1=1, POS2=0):**

- P4-11 = [jumlah putaran]
- P4-12 = [jumlah pulsa dalam putaran]
- P4-13 = [kecepatan dalam r/min]

**8. Posisi 5-8:** Lanjutkan dengan parameter P4-14 hingga P4-25 untuk posisi 5-8 jika diperlukan.

## **Contoh Pengaturan Posisi:**

Untuk pengaturan sederhana dengan 4 posisi, Anda bisa menggunakan:

**Posisi 1:** Putar 1 putaran dengan kecepatan 500 rpm

- P4-2 = 1 (1 putaran)
- P4-3 = 0 (0 pulsa tambahan)
- P4-4 = 500 (500 rpm)

**Posisi 2:** Putar 2 putaran dengan kecepatan 1000 rpm

- P4-5 = 2 (2 putaran)
- P4-6 = 0 (0 pulsa tambahan)
- P4-7 = 1000 (1000 rpm)

**Posisi 3:** Putar -1 putaran (arah berlawanan) dengan kecepatan 800 rpm

- P4-8 = -1 (-1 putaran)
- P4-9 = 0 (0 pulsa tambahan)
- P4-10 = 800 (800 rpm)

**Posisi 4:** Kembali ke posisi awal (0) dengan kecepatan 1500 rpm

- P4-11 = 0 (0 putaran)
- P4-12 = 0 (0 pulsa tambahan)
- P4-13 = 1500 (1500 rpm)

## **Pemrograman Arduino**

### **Kode Dasar untuk Arduino**



```

// Program Kontrol AC Servo Drive P100S dengan Arduino Uno
// Mode kontrol: Internal Position

// Definisi pin
const int CTRG_PIN = 2;    // DI1 (Pin 20) - Trigger
const int POS0_PIN = 3;    // DI2 (Pin 4) - Bit 0 pemilihan posisi
const int POS1_PIN = 4;    // DI3 (Pin 19) - Bit 1 pemilihan posisi
const int POS2_PIN = 5;    // DI4 (Pin 3) - Bit 2 pemilihan posisi
const int ZSP_PIN = 6;     // D04 (Pin 37) - Zero Speed Signal

// Delay untuk menunggu motor berhenti
const int MOTOR_DELAY = 1000; // ms

void setup() {
    // Inisialisasi pin
    pinMode(CTRG_PIN, OUTPUT);
    pinMode(POS0_PIN, OUTPUT);
    pinMode(POS1_PIN, OUTPUT);
    pinMode(POS2_PIN, OUTPUT);
    pinMode(ZSP_PIN, INPUT);

    // Inisialisasi status
    digitalWrite(CTRG_PIN, LOW);
    digitalWrite(POS0_PIN, LOW);
    digitalWrite(POS1_PIN, LOW);
    digitalWrite(POS2_PIN, LOW);

    Serial.begin(9600);
    Serial.println("Kontrol AC Servo Drive P100S");
    Serial.println("1-4: Pilih posisi");
}

void loop() {
    if (Serial.available() > 0) {
        char cmd = Serial.read();

        if (cmd >= '1' && cmd <= '8') {
            int position = cmd - '0'; // Konversi karakter ke angka
            moveToPosition(position);
        }
    }
}

```

```

// Fungsi untuk memilih dan menggerakkan ke posisi tertentu
void moveToPosition(int position) {
    Serial.print("Bergerak ke posisi: ");
    Serial.println(position);

    // Tunggu jika motor masih bergerak
    waitForMotorStop();

    // Pilih posisi dengan mengatur POS0, POS1, POS2
    // Posisi 1: 000, Posisi 2: 001, Posisi 3: 010, dll.
    digitalWrite(POS0_PIN, (position - 1) & 0x01); // Bit 0
    digitalWrite(POS1_PIN, ((position - 1) >> 1) & 0x01); // Bit 1
    digitalWrite(POS2_PIN, ((position - 1) >> 2) & 0x01); // Bit 2

    // Tunda sebentar untuk memastikan sinyal POS stabil
    delay(50);

    // Trigger pergerakan dengan memberikan pulsa pada CTRG
    digitalWrite(CTRG_PIN, HIGH);
    delay(50);
    digitalWrite(CTRG_PIN, LOW);

    Serial.println("Perintah dikirim");
}

// Fungsi untuk menunggu motor berhenti
void waitForMotorStop() {
    // Tunggu hingga sinyal ZSP HIGH (motor dalam keadaan diam)
    int timeout = 10000; // Timeout 10 detik
    int startTime = millis();

    Serial.println("Menunggu motor berhenti...");

    while (digitalRead(ZSP_PIN) != HIGH) {
        delay(10);
        if (millis() - startTime > timeout) {
            Serial.println("Timeout menunggu motor berhenti");
            break;
        }
    }
}

Serial.println("Motor berhenti");
}

```



## Penjelasan Kode:

1. **Definisi Pin:** Mengatur pin Arduino untuk terhubung ke servo drive.
2. **Setup:** Mengatur pinMode dan kondisi awal.
3. **Loop:** Membaca perintah dari Serial Monitor.
4. **moveToPosition:** Fungsi utama untuk memilih dan menggerakkan motor ke posisi tertentu.
  - Mengatur bit POS0, POS1, POS2 sesuai posisi yang dipilih
  - Memberikan pulsa pada CTRG untuk memicu pergerakan
5. **waitForMotorStop:** Fungsi untuk menunggu motor berhenti (sinyal ZSP HIGH).

## Pengujian Sistem

1. **Upload Kode:** Upload kode Arduino ke board Arduino Uno.
2. **Buka Serial Monitor:** Buka Serial Monitor di Arduino IDE (baud rate 9600).
3. **Uji Kontrol:** Ketik angka 1-8 di Serial Monitor untuk menggerakkan motor ke posisi yang sudah diprogram.
4. **Verifikasi Pergerakan:** Pastikan motor bergerak ke posisi yang benar.

## Troubleshooting

### Masalah Umum dan Solusi:

1. **Motor Tidak Bergerak:**
  - Periksa parameter servo drive (PA4, PA14)
  - Pastikan servo enabled (parameter PA53=1 atau hubungkan DI dengan fungsi SON ke +24V)
  - Periksa kabel dan koneksi
  - Pastikan tidak ada alarm aktif (lihat kode error di display servo)
2. **Gerakan Tidak Akurat:**
  - Sesuaikan parameter gain (PA9, PA5)
  - Periksa pengaturan posisi di parameter P4
3. **Motor Bergetar atau Tidak Stabil:**
  - Kurangi kecepatan (parameter P4-4, P4-7, dll)
  - Sesuaikan parameter filter (PA7, PA8)
4. **Alarm pada Servo Drive:**
  - Lihat kode error di Chapter 8 (manual servo)

- Reset alarm dengan parameter PA61=1 atau lewat pin ARST

#### 5. **Arduino Tidak Mengendalikan Servo:**

- Periksa kembali koneksi
- Pastikan level tegangan sesuai (gunakan optokopler jika diperlukan)
- Verifikasi parameter I/O (P3-0 hingga P3-3)

## **Tips dan Saran**

#### 1. **Keamanan Selalu Utama:**

- Pastikan ada tombol emergency stop yang terpisah
- Jangan menyentuh terminal saat servo drive terhubung ke listrik
- Perhatikan kabel ground untuk mencegah kejutan listrik

#### 2. **Memulai dengan Mudah:**

- Mulai dengan kecepatan rendah untuk pengujian
- Gunakan sedikit posisi (2-4) sebelum menggunakan semua 8 posisi
- Uji sistem tanpa beban sebelum menghubungkan beban mekanis

#### 3. **Peningkatan Sistem:**

- Tambahkan limit switch untuk keamanan tambahan
- Gunakan LCD untuk menampilkan status servo
- Implementasikan komunikasi serial untuk monitoring yang lebih detail

---

Semoga panduan ini membantu Anda memahami dan mengimplementasikan kontrol AC Servo Drive P100S menggunakan Arduino Uno. Jika Anda memiliki pertanyaan lebih lanjut atau membutuhkan bantuan tambahan, jangan ragu untuk bertanya.