

1. Apakah perbedaan open-loop system dan close-loop system?

Singkatnya, open-loop system adalah sistem perulangan yang tidak peduli dengan apa output yang dihasilkan, ia hanya peduli dengan input yang diberikan. Sedangkan close-loop system peduli dengan apa output yang dihasilkan sehingga dapat mengoreksi proses yang ada di dalam loop tersebut sesuai dengan output yang dihasilkan (apakah sesuai atau tidak).

2. Apakah sistem yang anda kerjakan dan simulasikan di atas termasuk open-loop system atau close-loop system? Justifikasi jawaban anda!

Sistem yang Saya kerjakan menggunakan keduanya, closed-loop dan open-loop. closed-loop yang saya gunakan terdapat pada sistem MPU6050 yang menerima input dari sensor dan menggerakkan servo tiap terjadi perubahan input. sedangkan pada PIR, ia hanya menerima satu kali input lalu membuat semua servo bergerak pada suatu posisi dan tidak peduli dengan outputnya.

3. Jelaskan fungsi masing-masing sensor yang digunakan pada sistem di atas!

Sensor MPU6050 sebenarnya mendeteksi 2 gerakan, yaitu akselerasi dan rotasi yang dialami sensor. pada akselerometer, sensor mendeteksi gaya gravitasi yang dialaminya untuk memperoleh data sudut kemiringan, roll dan pitch. Sebaliknya, giroskop mendeteksi perubahan rotasi yang dialami sensor dan digunakan untuk mendapatkan yaw. Sedangkan untuk PIR (Passive Infrared), sensor ini sebenarnya tidak memancarkan apapun, ia hanya mendeteksi gelombang inframerah yang ada di sekitarnya. Ketika ada benda panas yang terdeteksi di sekitarnya, seperti tangan manusia atau hewan, sensor akan mengirimkan sinyal HIGH pada mikrokontroler.

4. Jelaskan alasan, fungsi, dan arah tuju koneksi setiap pin ESP32 yang dimanfaatkan dari skematik (poin 5 Informasi Pengerjaan) yang telah anda buat!

Saya menggunakan protokol I2C pada MPU6050 untuk berkomunikasi dengan ESP32, Saya menghubungkan GPIO 21 kepada pin SDA dan GPIO 22 pada pin SCL di MPU. Tujuannya adalah agar data bisa saling dikirim kepada keduanya secara terkontrol (oleh SCL), serta alasannya, GPIO 21 & 22 pada ESP32 adalah pin SDA & SCL standar untuk I2C. Pada pin PIR, Saya menghubungkannya ke GPIO 13 agar sensor PIR dapat mengirimkan sinyal HIGH kepada ESP32 dan membuatnya bekerja dengan baik. Alasannya, GPIO 13 adalah pin digital yang umum yang dapat digunakan sebagai input kepada mikrokontroler. Pada output PWM, tujuannya untuk memberikan output (perintah) kepada servo. Alasannya, pin-pin yang saya pilih (GPIO 23, 5, 19, 18, dan 17) merupakan pin GPIO standar pada ESP32 yang bisa menghasilkan sinyal PWM dan jaraknya yang cukup dekat dari mikrokontroler sehingga kabel dapat tersusun lebih rapi. Terakhir adalah VCC dan GND, VCC berfungsi sebagai pemberi daya kepada komponen lain, sedangkan GND berfungsi untuk menyediakan jalur kembali bagi arus listrik untuk kembali ke ESP32.

- 5. Dalam suatu rapat monitoring, anda diminta untuk menjelaskan kode yang anda buat ke rekan kerja tim anda yang berbeda divisi dengan anda. Buatlah penjelasan yang mudah dipahami untuk menjelaskan alur bagaimana sistem yang anda program bekerja berdasarkan eksekusi kode yang telah dibuat hingga ke eksekusi yang dilakukan oleh mikrokontroler, sensor, dan aktuator yang ada!**

Program ini sebenarnya cukup sederhana, ide utamanya adalah sensor mendeteksi gerakan, lalu data diterima oleh mikrokontroler, kemudian data akan diproses oleh program sederhana, dan terakhir mikrokontroler akan memberikan perintah kepada servo sesuai dengan input yang diterima oleh sensor. Berikut adalah penjelasan lebih lengkapnya:

- Bagian awal program, hanya menuliskan definisi-definisi yang diperlukan agar kita tidak perlu mengingat hal-hal detail tapi penting, seperti nomor GPIO yang terhubung dengan masing-masing sensor, servo, dsb. Selain itu, bagian ini hanya melakukan import library (kode sederhana yang digunakan untuk mengakses hardware, library ini biasanya disediakan oleh produsen hardware).
- pada void setup(), bagian ini adalah function yang hanya dijalankan satu kali di awal yang bertujuan untuk melakukan setup pada semua komponen untuk memberi tahu bahwa “program sudah dimulai, bersiap-siaplah!”.
- Pada bagian void loop(), bagian ini juga merupakan fungsi, tapi fungsi ini akan dijalankan terus-menerus hingga program dimatikan. Bagian inilah yang merupakan logika utama program sederhana yang Saya katakan di awal.
- Pada bagian awal, program akan mengecek (membuat percabangan) apakah sensor PIR mendeteksi gerakan atau tidak? jika iya, maka semua servo akan bergerak secara serentak menuju suatu posisi, jika tidak maka akan masuk ke percabangan satunya.
- Ketika PIR tidak mendeteksi gerakan, program akan menggerakkan servo 1 & 2 berkebalikan dengan arah rotasi terhadap sumbu-x (roll) yang diterima oleh MPU, jika rotasi yang terdeteksi adalah 0 maka servo juga akan bergerak 0 derajat (yang artinya tidak bergerak). Kemudian, program akan menggerakkan servo 3 & 4 searah dengan pergerakan rotasi terhadap sumbu-y (pitch) yang dideteksi oleh MPU, sama seperti sebelumnya, jika rotasi yang terdeteksi adalah 0 (tidak ada pergerakan), maka servo 3 & 4 juga akan bergerak 0 derajat (tidak bergerak). setelah itu program akan masuk ke program khusus untuk servo 5 yang lebih rumit.
- Logika khusus untuk servo 5. karena servo 5 memiliki kondisi untuk ‘delay dan kembali’ (sesuai dengan soal yang diberikan), maka pada awal bagian ini, program akan mengecek apakah servo 5 sedang dalam masa ‘delay dan kembali’ atau tidak. Jika tidak, program akan mengecek lagi apakah sedang terjadi rotasi terhadap sumbu-z (yaw) atau tidak. jika masih terjadi pergerakan yaw, maka servo 5 akan mengikuti pergerakan rotasi itu. Jika tidak terjadi pergerakan, program akan mengecek ulang “apakah ini benar-benar stabil (tidak bergerak) atau hanya kesalahan hitung sesaat, karena perulangan terjadi sangat cepat.” maka di sinilah fungsi counter dalam program (bagi yang ingin memahami programnya). jika sudah dipastikan bahwa benar-benar tidak ada pergerakan yaw, maka servo 5 akan dimasukkan ke dalam kondisi ‘delay dan kembali’ dan meng-input variabel untuk waktu saat ini, yang pada akhirnya digunakan untuk menghitung apakah sudah delay 1 detik atau belum. Kembali ke percabangan awal pada logika untuk servo 5, yang menanyakan apakah servo 5 sedang dalam kondisi ‘diam dan kembali’ atau tidak. Jika iya, maka program akan mengecek, apakah sekarang sudah delay 1 detik atau belum (dengan

mengurang variabel yang menyimpan waktu terakhir stabil dengan waktu saat ini). Jika belum, program akan lanjut ke bagian berikutnya (loop awal, karena bagian ini adalah akhir dari program), jika sudah 1 detik, maka servo5 akan kembali ke posisi awal. Intinya, program khusus ini memastikan apakah yaw masih bergerak atau tidak, jika tidak maka ia akan masuk ke mode 'delay dan kembali' (kecuali ia memang sudah berada di posisi kembali), jika masih bergerak (atau masih berada di tempat kembali), maka lanjutkan bergerak.