Fungsi Program Arduino

|  |
| --- |
| 1. |
|  | pinMode() |
|  | Berfungsi untuk mengkonfigurasi pin tertentu agar berperilaku sebagai input atau output. |
|  |  |
|  |  |
|  |  |
|  | 2. digitalWrite() |
|  | Berfungsi untuk memberi nilai HIGH atau LOW ke pin digital. Jika pin telah dikonfigurasi sebagai OUTPUT dengan pinMode(), tegangan akan diatur ke nilai yang sesuai: 5V (atau 3.3V) untuk HIGH, 0V (ground) untuk LOW. |
|  | Jika pin dikonfigurasi sebagai INPUT, digitalWrite() akan mengaktifkan (HIGH) atau menonaktifkan (LOW) dengan pullup internal pada pin input. Disarankan dalam mengatur pinMode() INPUT\_PULLUP untuk mengaktifkan resistor pull-up internal. |
|  |  |
|  | 3. digitalRead() |
|  | Membaca nilai dari pin digital tertentu, baik HIGH atau LOW. |
|  |  |
|  | 4. analogWrite() |
|  | adalah fungsi yang digunakan untuk menuliskan suatu nilai berupa angka pada sebuah komponen, misalnya LED. Pengguna dapat mengatur seberapa terang cahaya dari lampu LED saat menyala, tergantung pada nilai yang dituliskan. Fungsi ini akan berguna ketika kita mulai bermain dengan sensor, di mana nilai yang terbaca seringkali berupa analog (memiliki banyak nilai, misal 0-1023), bukan digital (hanya memiliki 2 nilai, 0 (LOW) dan 1(HIGH)). Contoh:analogWrite(13, 1023) artinya Arduino akan memberikan instruksi nilai maksimal pada pin 13, sehingga komponen yang terkoneksi di pin 13 akan menyala maksimal (1023). |
|  |  |
|  | 5. analogRead(pin) |
|  |  |
|  |  |
|  | Fungsi ini mirip dengan fungsi digitalRead(), yaitu membaca nilai masukan pada suatu pin. Bedanya adalah fungsi analogRead() akan menghasilkan nilai dari 0 hingga 1023, yang merepresentasikan voltase 0 v hingga 5 v. Contoh:analogRead(0) artinya Arduino akan membaca nilai input dari pin 0. |
|  |  |
|  | 6. delay(time) |
|  | Fungsi ini digunakan untuk memberikan jeda antar fungsi. Nilai time adalah waktu lamanya jeda dalam satuan ms (milisekon), di mana 1 detik setara dengan 1.000 milisekon. |
|  |  |
|  | 7.Serial.begin(baudrate) |
|  |  |
|  |  |
|  | Pengguna dapat melakukan komunikasi serial antara Arduino dengan PC, dengan menggunakan Serial Monitor yang disediakan pada Arduino IDE. Pada Serial Monitor, kita bisa melihat data yang dikirim dari Arduino ke PC. Selain itu, kita juga bisa mengirim data ke Arduino dengan cara mengetikkannya pada textbox di bagian atas Arduino IDE. Untuk memakai serial, yang pertama harus dilakukan adalah melakukan inisiasi dengan menggunakan fungsi Serial.begin(baudrate). Variabel baudrate merupakan rasio modulasi, dan harus dicocokkan dengan baudrate hardware yang akan dikomunikasikan. Contoh: Serial.begin(9600)artinya komunikasi akan berjalan pada rasio modulasi/baudrate 9600. |
|  |  |
|  |  |
|  |  |
|  | 8.Serial.available() |
|  |  |
|  |  |
|  | Fungsi ini digunakan untuk mengetes apakah ada input data dari hardware yang disambungkan ke serial port, misalnya dari PC. Fungsi ini akan menghasilkan 1 apabila ada masukan, dan 0 apabila tidak ada masukan. |
|  |  |
|  |  |
|  |  |
|  | 9.Serial.read() |
|  |  |
|  |  |
|  | Fungsi ini berfungsi untuk membaca karakter pada serial port. Karakter yang dibaca akan disimpan dalam bentuk ASCII (misalnya karakter ‘0’ memiliki representasi ASCII yaitu 48). |
|  |  |
|  |  |
|  |  |
|  | 10.Serial.print() dan Serial.println() |
|  |  |
|  |  |
|  | Fungsi ini digunakan untuk menuliskan suatu kalimat ke Serial Monitor, tetapi tidak mengirimkan data apapun, alias hanya digunakan untuk memberikan teks visual pada pengguna. Serial.print("text") digunakan untuk menulis "text", sedangkan Serial.println("text") dipakai untuk menuliskan kata "text" dan diakhiri dengan enter (kalimat selanjutnya ada di baris berikutnya). |
|  |  |
|  |  |
|  |  |
|  | 11.Serial.write(VAL) |
|  |  |
|  |  |
|  | Untuk mengirimkan data dari arduino ke PC, kita bisa menggunakan fungsiSerial.write(VAL). Nilai VAL adalah data yang ingin dikirimkan dari arduino ke PC, dengan ukuran 1 byte. |
|  |  |
|  | 12.While Loop |
|  | while (expression) //selama expresi bernilai true laksanakan perulangan blok statemen dibawahnya. |
|  | { |
|  | statement1; |
|  | statement2; |
|  | ….. |
|  | } |
|  | atau jika satement hanya 1 baris statement tdk perlu di berikurung { } |
|  | while(expression) |
|  | statement; |
|  | 13.Do/While Loop |
|  | do //jalankan statement dibawah ( minimal 1 kali perulangan dilaksanakan) , |
|  | { |
|  | statement1; |
|  | statement2; |
|  | … |
|  | } |
|  | while (expression); // jika ekspresi true , jalankan kembali perulangan. |
|  | atau jika satement hanya 1 baris |
|  | do |
|  | statement; |
|  | while (expression); |
|  | 14.For Loop |
|  | for (expr1; expr2; expr3) //laksanakan perulangan sebanyak expr2 dimulai dari expr1. |
|  | { |
|  | statement1; |
|  | statement2; |
|  | … |
|  | } |
|  | atau jika satement hanya 1 baris |
|  | for(expr1; expr2; expr3) |
|  | statement; |
|  | 15.If/Else |
|  | if (expression) //jika ekspresi bernilai true laksanakan blok statemen dibawahnya |
|  | { |
|  | statement1; |
|  | statement2; |
|  | … |
|  | } |
|  | atau jika satement hanya 1 baris |
|  | if(expression) |
|  | statement; |
|  | 16.SWITCH/CASE |
|  | switch (expression) |
|  | { |
|  | case const1: // jika expression = const1 eksekusi staement2 dibawahnya |
|  | statement1; |
|  | statement2; |
|  | case const2: // jika expression = const2 eksekusi statemenibawahnya |
|  | statement3; |
|  | statement4; |
|  | case constX: // jika espression=constX eksekusi statemen dibawahnya |
|  | statement5; |
|  | statement6; |
|  | default: |
|  | statement7; // jika ekspression tdk sama dgn const1,cons2, dan constX |
|  | statement8; |
|  | } |
|  | 17.BREAK, CONTINUE,AND GOTO |
|  | break, continue, & goto statements digunakan untuk merubah alur eksekusi untuk statement pengulangan for, while, do/while, and switch . |
|  | Break |
|  | break statement digunakan untuk keluar dari pengulangan for, while, do/while, atau switch . break keluar dari blok pengulangan dimana dia berada saja. |
|  | Continue |
|  | continue akan menyebabkan program memulai iterasi berikutnya dari statemen pengulangan while, |
|  | do/while, atau for loop. continue sama seperti break pengulangan akan berhenti pada point yg dituju (di skip), bedanya continue akan memulai loop lagi, dari atas, sedangkan break memaksa keluar dari loop / pengulangan . |
|  | setup() |
|  | { |
|  | } |
|  | loop() |
|  | { |
|  | int i; |
|  | for(i; i<7; i++) |
|  | { |
|  | if(i==3) |
|  | continue; |
|  | printf(%d, i ) |
|  | } |
|  | } |
|  | 18.OPERATOR DAN EXPRESSI |
|  | Operator Penugasan ( “=” ) |
|  | Setelah variabel telah dideklarasikan, operasi dapat dilakukan pada variable menggunakan operator penugasan (=). Sebuah nilai yang diberikan untuk variabel dapat berupa sebuah konstanta, variabel, atau ekspresi. Sebuah ekspresi dalam bahasa C adalah kombinasi dari operan (pengidentifikasi) dan operator. operaor penugasan antara lain sebagai berikut: |
|  | Variable = nilai , |
|  | contoh nilai\_max = 10; |
|  | Operator Arithmetic : |
|  | Perkalian \* |
|  | Pembagian / |
|  | Modulo % |
|  | Penjumlahan + |
|  | Pengurangan atau Negasi – |
|  | 19.Operator Logika |
|  | AND && |
|  | OR || |
|  | Operator relasional |
|  | sama dengan == |
|  | tdk sama dengan != |
|  | lebih kecil < |
|  | lebih kecil atau sama dengan <= |
|  | lebih besar > |
|  | lebih besar atau sama dengan >= |
|  | increment |
|  | ++variable atau variable++ artinya variable=variable+1 , |
|  | misal ++X atau X++ artinya X = X+1 |
|  | perbedaaanya adalah untuk ++X , tambahkan dulu nilai X dgn 1 baru di proses baris kode tsb, sedangkan X++ , proses dulu baris code baru tambahkan nilai X dgn 1 |
|  | deccrement |
|  | –variable atau variable– artinya variable=variable-1 , |
|  | misal –Y atau Y– artinya Y = X-1 |
|  | perbedaaanya adalah untuk –Y , kurangi dulu nilai Y dgn 1 baru di proses baris kode tsb, sedangkan Y– , proses dulu baris code tsb baru kurangi nilai Y dgn 1 |
|  | Compound |
|  | compound digunakan hanya untuk menyingkat baris expresi saja contoh nya sbb |
|  | a += 3; artinya a = a + 3 |
|  | b -= 3; artinya b = b – 3 |
|  | c \*= 3; artinya c = c \* 3 |
|  | d /= a; artinya d = d / a |
|  | a |= 3; artinya a = a OR 3 |
|  | b &= 3; artinya b = b AND 3 |
|  | c ^= 3; artinya c = c ^ 3 |
|  | PORTD &= 3; artinya PORTD = PORTD & 3 |
|  | kondisonal Expressi |
|  | if(expression\_A) |
|  | expression\_B; |
|  | else |
|  | expression\_C; |
|  | baris kode diatas bisa digantikan oleh kondisional expressi: |
|  | expression\_A ? expression\_B : expression\_C; |
|  | contoh : (A > 10) ? x = 5 : x =22; |
|  | kondisional expresi diatas artinya sama dengan : |
|  | IF(A>10) |
|  | { |
|  | x=5; |
|  | } |
|  | Else |
|  | { |
|  | x= 22 |
|  | } |
|  | .. |
|  | 20.FUNGSI |
|  | yang dimaksud fungsi adalah sebuah blok statement yg melaksanakan tugas tertentu dan bisa |
|  | dipakai lebih dari 1 kali di dlm program. |
|  | program bahasa C terdiri dari fungsi fungsi dan sebuah program c minimal terdiri dari 1 fungsi . |
|  | fungsi setup() dijalankan pertama kali saat program dijalankan. |
|  | fungsi terdiri dari dua bagian utama yaitu kepala fungsi dan blok fungsi |
|  | berikut ini bentuk umum sebuah fungsi adalah : |
|  | type name\_fungsi (type paramameter) |
|  | { |
|  | statement |
|  | } |
|  | kepala fungsi terdiri dari type return value, nama fungsi , types dan nama parameter(jika ada) . |
|  | Statement di block fungsi menyatakan apa yg fungsi harus lakukan. |
|  | jika return value atau parameter tdk ada maka typennya ditulis void seperti dibawah ini |
|  | void namefungsi (void) |
|  | { |
|  | statement |
|  | } |
|  | Structures, Union, dan Data Storage |
|  | 21.Digital Input |
|  | void pinMode (int pin, int mode) |
|  | memerintahkan Arduino apakah pin akan diset sbg digital input atau output. Mode dapat berupa Input atau OUTPUT. |
|  | int digitalRead (int pin) |
|  | Membaca nilai dari pin digital. Nilai yg didapat dapat berupa 1 atau 0. |
|  | void digitalWrite (int pin, int value) |
|  | Menulis nilai ke pin digital. Nilai yg ditulis dapat berupa 1 atau 0. |
|  | Analog Input |
|  | int analogRead (int pin) |
|  | Membaca nilai dari pin analog. Nilai yg didapat adalah antara 0 dan 1023. |
|  | void analogWrite (int pin, int value) |
|  | Menulis nilai analog ke pin. nilai yg ditulis antara 0 dan 255. |