|  |  |  |
| --- | --- | --- |
| **KELOMPOK** |  | **ACC** |
|  |  |  |
|  | Kurniawan Aji S (D400150077) |  |
|  | Umar Abdul Aziz (D400150007) | **Tanggal ACC :** |
|  | Khairul Fikri (D400154001) | **Revisi Tanggal :** |

**LAPORAN TUGAS BESAR**

Multitasking Kendali LED menggunakan Sensor IR Obstacle dan Motor DC Menggunakan Sensor Ultrasonik

1. **TUJUAN**
2. Praktikan dapat memahami salah satu sistem operasi yang ada pada Arduino yaitu FreeRTOS.
3. Mahasiswa dapat memahami tentang pengendalian 2 aktuator melalui 2 sensor dengan proses multitasking
4. Sebagai syarat tugas besar Praktikum Sistem Operasi Mikroprosessor
5. **ALAT DAN BAHAN**
6. Laptop / PC yang telah diinstal software Arduino
7. Arduino Uno
8. LED
9. Sensor IR Obstacle
10. Motor DC
11. Sensor Ultrassonik
12. Kabel Jumper
13. Project Board
14. **DASAR TEORI**
15. **FreeRTOS**

FreeRTOS (Free Real-time operating systems) adalah sebuah *real time operating system* pada Arduino. FreeRTOS ini banyak digunakan oleh mikrokontroler untuk kebutuhan sistem operasinya. FreeRTOS didistribusikan secara gratis lewat GPL dengan beberapa pengecualian untuk kalangan industri tertentu. RTOS (Real Time Operating System) adalah sistem operasi yang digunakan untuk memenuhi kebutuhan aplikasi yang bersifat real time. Real time disini berarti ia membutuhkan waktu kinerja setiap saat dimana ia dibutuhkan saat itu juga. Salah satu kunci dari keberhasilan RTOS adalah kemampuannya untuk melakukan kerja secara konsisten baik secara waktu yang ia butuhkan maupun secara task aplikasi yang mampu ia kerjakan.

Dalam FreeRTOS, itu sedikit berbeda. FreeRTOS dirancang khusus untuk dijalankan pada aplikasi dengan waktu yang sangat tepat dan tingkat keandalan yang tinggi. Jadi, RTOS digunakan ketika kendala sementara dalam suatu sistem sangat penting. Ada yang berbeda jenis RTOS, Soft, Firm dan Hard. Sebelum masuk ke klasifiksi RTOS harus memperkenalkan terlebih dahulu mengenai jitter. Jitter adalah ukuran seberapa banyak waktu eksekusi dengan jadwal yang ditentukan atas iterasi selanjutnya. Sistem operasi real-time dioptimalkan meminimalkan jitter.

• Soft RTOS ditandai dengan jitter tinggi, Soft RTOS digunakanketika kendala waktu di mana tugas-tugas harus dijalankan kurang penting.Biasanya digunakan dalam sistem ketika risiko bahwa tugas tidak dijalankan pada waktunya tidak kritis.

• Hard RTOS yang keras memiliki lebih sedikit jitter daripada Soft RTOS, dalam jenis RTOS ini kendala waktu sangat penting dan jika tugas tidak dijalankan dalam waktu, hasilnya tidak lagi berguna dan sistem tidak berfungsi lagi.

• Firm RTOS terletak antara Soft RTOS dan Hard RTOS. Jumlah Jitternya kurang dari Soft RTOS dan lebih tinggi daripada Hard RTOS.

1. **LED**

LED adalah singkatan dari “Light Emitting Diode”. Yang berarti LED adalah perangkat semi-konduktor yang menghasilkan cahaya ketika arus listrik melewati celah antara katoda dan anoda didalam sistem perangkat tsb. LED juga disebut “Solid State Lighting” karena chip LED disolder ke Printed Circuit Board (PCB) dan oleh karena itu tidak memiliki artikel-artikel yang longgar / filamen seperti bola lampu pijar, atau zat beracun seperti gas merkuri pada Lampu Hemat Energy (LHE).

LED telah beredar cukup lama, tetapi baru akhir-akhir ini produsen-produsen LED telah mulai memperluas lini produk mereka dari lampu dioda sederhana, yang digunakan terutama untuk lampu indikator, ke versi yang lebih canggih, lebih efisien dan lebih terang. Dalam dekade terakhir ini, kemajuan teknologi LED telah berhasil membuat LED yang lebih terang dan berdaya lebih tinggi untuk diaplikasikan ke dalam senter, lampu outdoor, lampu mobil, dan sebagainya. Dan hanya baru-baru ini, LED mulai digunakan secara luas untuk keperluan penerangan umum (penerangan dalam/luar ruangan, penerangan komersial, lampu dekorasi, lampu sorot, lampu panggung, dll).

1. **Sensor IR Obstacle**

Sensor IR terdiri dari IR LED dan Photodiode IR; bersama-sama mereka disebut sebagai Photo-Coupler atau Opto-Coupler. Seperti yang dikatakan sebelumnya, Sensor Hambatan Inframerah memiliki built-in pemancar IR dan penerima IR. Infrared Transmitter adalah dioda pemancar cahaya (LED) yang memancarkan radiasi infra merah. Oleh karena itu, mereka disebut IR LED. Meskipun LED IR tampak seperti LED normal, radiasi yang dipancarkan olehnya tidak terlihat oleh mata manusia. Penerima inframerah juga disebut sebagai sensor inframerah karena mereka mendeteksi radiasi dari pemancar IR. Penerima IR datang dalam bentuk fotodioda dan phototransistors. Photodioda inframerah berbeda dari dioda foto normal karena mereka hanya mendeteksi radiasi infra merah. Ketika pemancar IR memancarkan radiasi, ia mencapai objek dan beberapa radiasi memantulkan kembali ke penerima IR. Berdasarkan intensitas penerimaan oleh penerima IR, output dari sensor ditentukan.

1. **Sensor Ultrasonik**

Sensor ultrasonik adalah sebuah sensor yang memiliki fungsi untuk mengubah besaran fisis  alias bunyi menjadi besaran listrik, begitupun sebaliknya. Prinsip kerja sensor ultrasonik ini cukup simpel, yakni berdasarkan pantulan suatu gelombang suara sehingga dapat digunakan untuk mendefiniskan eksistensi atau jarak suatu benda dengan frekuensi tertentu.

Mengapa disebut sensor ultrasonik? Karena sensor ini menggunakan gelombang ultrasonik. Gelombang ultrasonik sendiri memiliki frekuensi yang sangat tinggi, mencapai 20.000 Hz yang tidak bisa didengar oleh telinga manusia. Bunyi dengan frekwensi setinggi itu hanya bisa didengar oleh hewan-hewan tertentu seperti kucing, anjing, kelelawar, sampai dengan lumba-lumba.

Bunyi dari sensor ultrasonik sendiri dapat merambat melalui benda padat, cair, atau gas. Namun yang paling bagus adalah benda cair. Tak heran jika sensor yang satu ini banyak diaplikasikan pada kapal selam dan alat-alat khusus untuk mengukur kedalaman air laut. Sayangnya bunyi sensor ultrasonik dapat diserap oleh benda-benda tekstil dan busa.

1. **Motor DC**

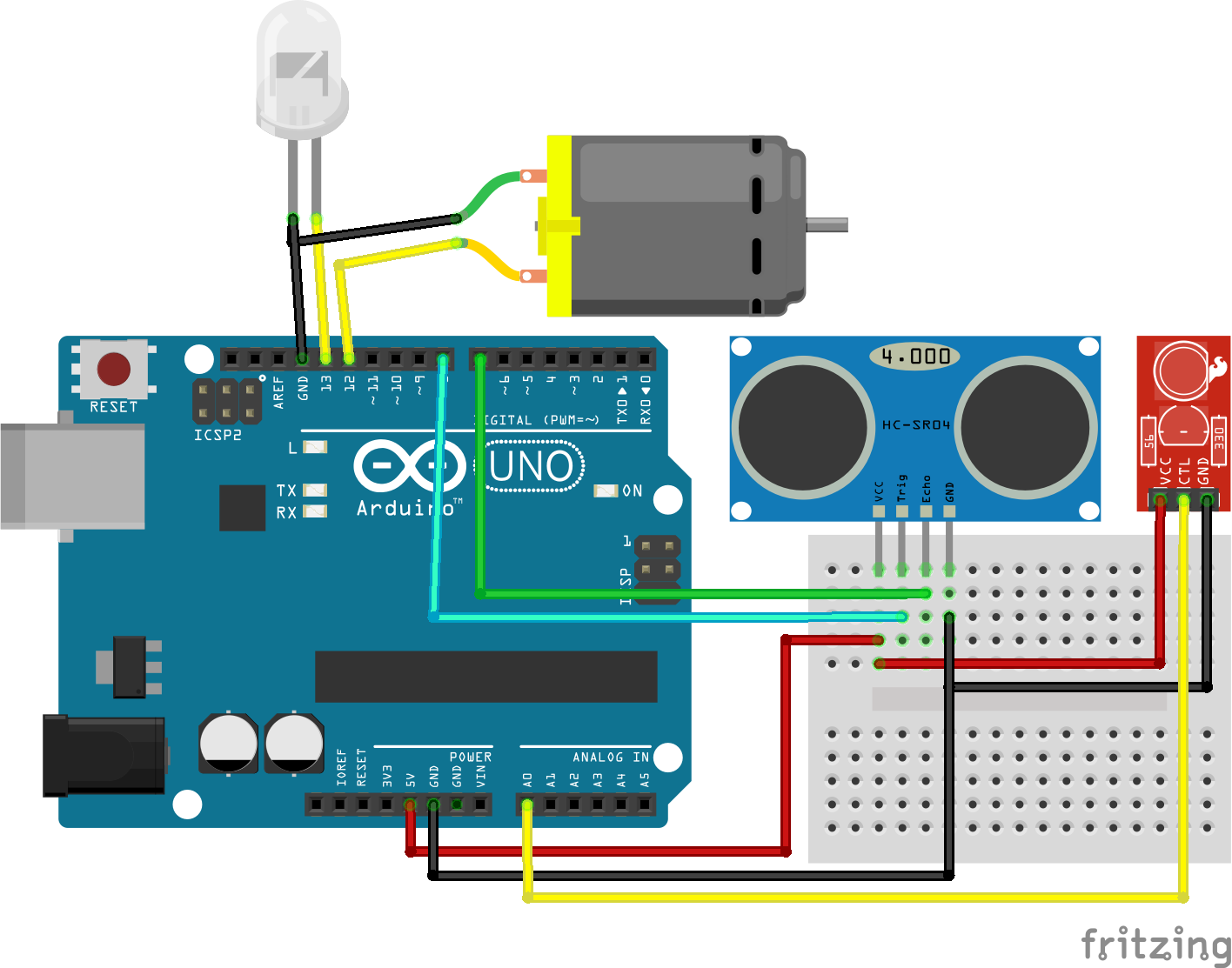
Motor Listrik DC atau DC Motor ini menghasilkan sejumlah putaran per menit atau biasanya dikenal dengan istilah RPM (Revolutions per minute) dan dapat dibuat berputar searah jarum jam maupun berlawanan arah jarum jam apabila polaritas listrik yang diberikan pada Motor DC tersebut dibalikan. Motor Listrik DC tersedia dalam berbagai ukuran rpm dan bentuk. Kebanyakan Motor Listrik DC memberikan kecepatan rotasi  sekitar 3000 rpm hingga 8000 rpm dengan tegangan operasional dari 1,5V hingga 24V. Apabile tegangan yang diberikan ke Motor Listrik DC lebih rendah dari tegangan operasionalnya maka akan dapat memperlambat rotasi motor DC tersebut sedangkan tegangan yang lebih tinggi dari tegangan operasional akan membuat rotasi motor DC menjadi lebih cepat. Namun ketika tegangan yang diberikan ke Motor DC tersebut turun menjadi dibawah 50% dari tegangan operasional yang ditentukan maka Motor DC tersebut tidak dapat berputar atau terhenti. Sebaliknya, jika tegangan yang diberikan ke Motor DC tersebut lebih tinggi sekitar 30% dari tegangan operasional yang ditentukan, maka motor DC tersebut akan menjadi sangat panas dan akhirnya akan menjadi rusak.

Pada saat Motor listrik DC berputar tanpa beban, hanya sedikit arus listrik atau daya yang digunakannya, namun pada saat diberikan beban, jumlah arus yang digunakan akan meningkat hingga ratusan persen bahkan hingga 1000% atau lebih (tergantung jenis beban yang diberikan). Oleh karena itu, produsen Motor DC biasanya akan mencantumkan Stall Current pada Motor DC. Stall Current adalah arus pada saat poros motor berhenti karena mengalami beban maksimal.

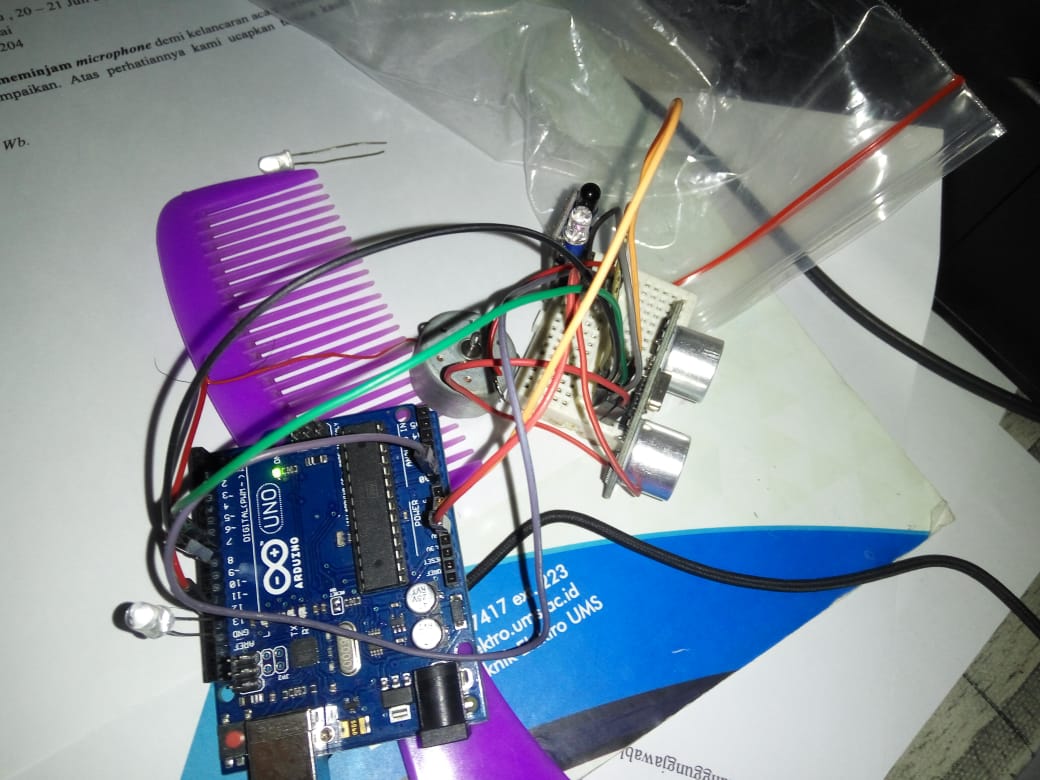
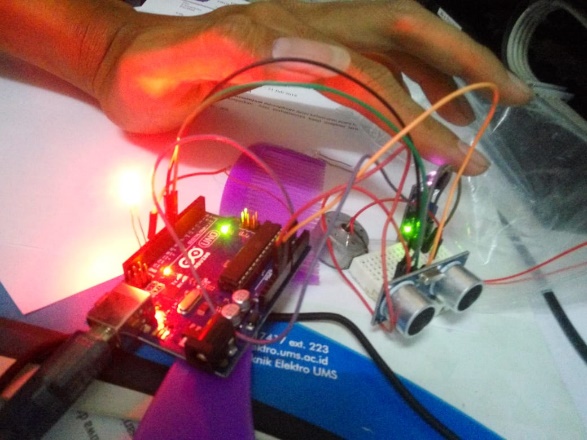
1. **HASIL PRAKTIKUM**

D.1 Hasil Percobaan 1

D.1.1 Skema Rangkaian



D.1.2 Foto Rangkaian

Sebelum Program Diupload Setelah Program Diupload

D.1.3 Script Program

#include <Arduino\_FreeRTOS.h>

#include <semphr.h>

int maximumRange = 200;

int minimumRange = 1 ;

long duration, distance;

#define echoPin 7 //Echo Pin

#define trigPin 8 //Trigger Pin

#define motorPin 12 //Motor Pin

#define dataPin A0 //Output Pin Sensor IR

#define ledPin 13 //Led Pin

SemaphoreHandle\_t xSerialSemaphore;

void SensorIR ( void \*pvParameters );

void SensorUltrasonik ( void \*pvParameters );

// the setup function runs once when you press reset or power the board

void setup() {

// initialize serial communication at 9600 bits per second:

Serial.begin(9600);

while (!Serial) {

; // wait for serial port to connect. Needed for native USB, on LEONARDO, MICRO, YUN, and other 32u4 based boards.

}

if ( xSerialSemaphore == NULL ){

xSerialSemaphore = xSemaphoreCreateMutex();

if (( xSerialSemaphore ) != NULL )

xSemaphoreGive(( xSerialSemaphore ));

}

// Now set up two tasks to run independently.

xTaskCreate(

SensorIR

, (const portCHAR \*)"Blink" // A name just for humans

, 128 // This stack size can be checked & adjusted by reading the Stack Highwater

, NULL

, 2 // Priority, with 3 (configMAX\_PRIORITIES - 1) being the highest, and 0 being the lowest.

, NULL );

xTaskCreate(

SensorUltrasonik

, (const portCHAR \*) "AnalogRead"

, 128 // Stack size

, NULL

, 1 // Priority

, NULL );

// Now the task scheduler, which takes over control of scheduling individual tasks, is automatically started.

}

void loop()

{

// Empty. Things are done in Tasks.

}

/\*--------------------------------------------------\*/

/\*---------------------- Tasks ---------------------\*/

/\*--------------------------------------------------\*/

void SensorIR( void \*pvParameters \_\_attribute\_\_((unused)) ) // This is a task.

{

//(void) pvParameters;

pinMode(dataPin, INPUT);

pinMode(ledPin, OUTPUT);

for (;;) // A Task shall never return or exit.

{

if ( xSemaphoreTake( xSerialSemaphore, ( TickType\_t ) 5 ) == pdTRUE ){

Serial.print("Nilai Sensor IR : ");

Serial.println(analogRead(A0));

vTaskDelay(50);

if(analogRead(A0) < 400)

digitalWrite(13,HIGH);

else

digitalWrite(13,LOW);

xSemaphoreGive ( xSerialSemaphore );

}

vTaskDelay(1);

}

}

void SensorUltrasonik( void \*pvParameters \_\_attribute\_\_((unused)) ) // This is a task.

{

//(void) pvParameters;

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

pinMode(motorPin, OUTPUT);

for (;;)

{

if ( xSemaphoreTake( xSerialSemaphore, ( TickType\_t ) 5 ) == pdTRUE )

{

digitalWrite(trigPin, LOW);

vTaskDelay(5);

digitalWrite(trigPin, HIGH);

vTaskDelay(15);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distance = duration/58.2;

Serial.print("Jarak Sensor Ultrasonik : ");

Serial.println(distance);

digitalWrite(motorPin, LOW);

if (distance > 30){

digitalWrite(motorPin, HIGH);}

xSemaphoreGive( xSerialSemaphore );

}

vTaskDelay(1);

}

}

void SensorUltrasonik( void \*pvParameters \_\_attribute\_\_((unused)) ) // This is a task.

{

//(void) pvParameters;

pinMode(8, OUTPUT);

pinMode(7, INPUT);

pinMode(12, OUTPUT);

for (;;)

{

if ( xSemaphoreTake( xSerialSemaphore, ( TickType\_t ) 5 ) == pdTRUE )

{

digitalWrite(8, LOW);

vTaskDelay(5);

digitalWrite(8, HIGH);

vTaskDelay(15);

digitalWrite(8, LOW);

duration = pulseIn(7, HIGH);

distance = duration/58.2;

Serial.print("Jarak Sensor Ultrasonik : ");

Serial.println(distance);

digitalWrite(12, LOW);

if (distance > 30){

digitalWrite(12, HIGH);}

xSemaphoreGive( xSerialSemaphore );

}

vTaskDelay(1);

}

}

1. **ANALISA**

Pada Tugas Besar ini menggunakan 1 buah Arduino Uno, 1 Buah LED, 1 buah sensor IR Obstacle, 1 buah motor DC dan 1 buah Sensor Ultrasonik. Kemudian LED sebagai output dihubungkan pada pin 13 yang dikendalikan oleh analog Read sensor IR Obstacle melalui pin A0 sebagai Input. Seperti hal nya LED, motor DC juga dihubungkan pada pin 12 sebagai output yang dikendalikan oleh sensor Ultrasonik melalui pin 7 dan 8 sebagai input. Kedua sensor dan actuator tersebut akan berkerja secara bersamaan ketika program sudah dinyalakan (multitasking).

Pada program Arduino dengan menggunakan sistem operasi FreeRTOS percobaan ini menggunakan 2 Task, untuk memudahkan penyalaan warna lampu sesuai yang diinginkan. Pada perintah void SensorIR( void \*pvParameters ); void setup() {} digunakan untuk mendaftarkan aplikasi atau task dalam sistem. Pada perintah xTaskCreate() digunakan untuk proses registrasi aplikasi dalam RTOS. Kemudian pada perintah void SensorIR(){} digunakan untuk membaca data dari sensor IR Obstacle dan LED akan menyala pada output LED jika data yang terbaca kurang dari 400 dan sebaliknya. Kemudian void SensorUltrasonik(){} digunakan untuk membaca data berupa jarak (cm) dan motor akan menyala jika data yang terbaca lebih dari 30 cm

1. **KESIMPULAN**
2. FreeRTOS (Free Real-time operating systems) adalah sebuah *real time operating system* pada Arduino yang banyak digunakan oleh mikrokontroler untuk kebutuhan sistem operasinya.
3. Pada sistem Analog dan digital Read ini menggunakan RTOS multitasking dengan menggunakan sensor IR Obstacle dan Sensor Ultrasonik dalam waktu yang sama untuk menghidupkan LED dan motor DC yang berfungsi sebagai aktuator.
4. Sistem operasi RTOS dalam program dibagi dalam beberapa task yang dapat diatur urutannya sesuai dengan kebutuhan.
5. Setiap task dapat disisipkan perintah loop ataupun tidak disisipkan sesuai dengan kebutuhan.