Modul 2 Pengenalan RTOS Arduino

A. Tujuan

1.Mahasiswa mampu menggunakan library free RTOS pada arduino

2.Mahasiswa dapat membandingkan & menganalisa perbandingan pemrograman arduino berbasiskan *void loop () & void setup ()* dengan menggunakan library *free RTOS arduino.*

B. Alat & Bahan

1. Arduino UNO
2. Kit Sensor & Aktuator
3. Project board & jumper
4. Library Free RTOS arduino

C. Landasan teori

What is a RTOS?

Before seeing what a real-time OS is, let's explain the di!erence between a general purpose operating system and a real-time operating system. Windows, Mac OS and Linux are general purpose OS, they are designed to run multiple applications and to give the illusion to the user that they are all running simultaneously. They are named as non-deterministic OS, in other words, the temporal constraint is not the most important factor.

In real-time OS, it's a bit di!erent. Real-time OS are specially designed to run applications with very precise timing and a high degree of reliability [1]. Thus, RTOS are used when the temporal constraint in a system is very important.

There are di!erent kinds of RTOS, *soft*, *firm* and *hard*. In order to characterize them, we will introduce the *jitter*. The Jitter is a measure of how much the execution time of a task di!ers over subsequent iterations. Real-time operating systems are optimized to minimize the jitter.

• *Soft real-time OS* are characterized with a high jitter, soft real-time OS are used when the time constraint in which tasks must be executed is less important. Typically used in systems when the risk that a task is not executed in time is not critical.

• *Hard real-time OS* have a less jitter than soft real-time OS, in this kind of RTOS the time constraint is very important and if a task is not executed in time, the result is no more useful and the system doesn't work anymore. Hard real-time OS are used, for example, in ABS (anti-lock braking system) or Airbags.

• *Firm real-time OS* are located between soft and hard real-time OS. The Jitter is less than for soft RTOS and higher than for hard RTOS.

[1, 2, 3, 4]

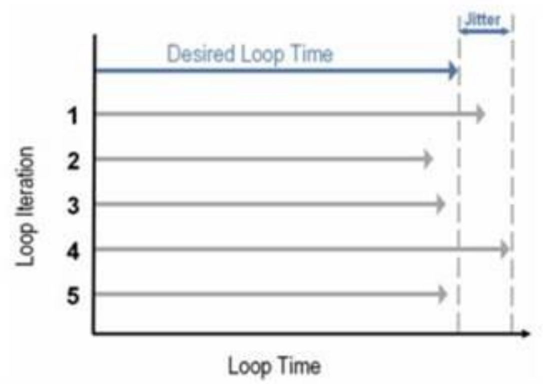

Figure 1: Illustration of the jitter

We can see clearly now, the di!erence between a real-time OS and a general purpose OS. This di!erence is the temporal constraint and reliability._*Nicolas Pantano Report*

D. Langkah kerja percobaan & pengamatan

D.1 Percobaan 1 blink led & analog read (see. Example library)

```
#include <Arduino_FreeRTOS.h>
void TaskBlink( void *pvParameters );
void TaskAnalogRead( void *pvParameters );
void setup() {
  Serial.begin(9600);
  while (!Serial) {
  }
  // Now set up two tasks to run independently.
  xTaskCreate(
    TaskBlink
    ,  (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(
    TaskAnalogRead
    ,  (const portCHAR *) "AnalogRead"
    ,  128  // Stack size
    ,  NULL
    ,  1  // Priority
    ,  NULL );
}
void loop()
{
}
void TaskBlink(void *pvParameters)  // This is a task.
{
  (void) pvParameters;
  pinMode(LED_BUILTIN, OUTPUT);
  for (;;) // A Task shall never return or exit.
  {
    digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
    vTaskDelay( 1000 / portTICK_PERIOD_MS ); // wait for one second
    digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
    vTaskDelay( 1000 / portTICK_PERIOD_MS ); // wait for one second
  }
}

void TaskAnalogRead(void *pvParameters)  // This is a task.
{
  (void) pvParameters;
  for (;;)
  {
    // read the input on analog pin 0:
    int sensorValue = analogRead(A0);
    // print out the value you read:
    Serial.println(sensorValue);
    vTaskDelay(1);  // one tick delay (15ms) in between reads for stability
  }
}
```

D.2 Percobaan 2

Pada percobaan ini buatlah beberapa task, kembangkan dari sketch percobaan 1

Tips : ingatlah cara pembuatan task hingga fungsi.

- Anda harus memasukan library RTOS pada IDE arduino #include <Arduino_FreeRTOS.h>
- Definisikan task anda, misal : void TaskDigitalRead(void *pvParameters);
- Kemudian buatlah task anda misal :

```
xTaskCreate(
    TaskDigitalRead
    ,  (const portCHAR *)"DigitalRead"  // 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 );
```

- Membuat fungsi task

```
void TaskBlink(void *pvParameters)  // This is a task.
{
  (void) pvParameters;

pinMode(LED_BUILTIN, OUTPUT);

  for (;;)                                    // A Task shall never return or exit.
  {
    digitalWrite(LED_BUILTIN, HIGH);
    vTaskDelay( 1000 / portTICK_PERIOD_MS );
    digitalWrite(LED_BUILTIN, LOW);
    vTaskDelay( 1000 / portTICK_PERIOD_MS );
  }
}
```

Tabel 2.1 Tabel Fungsi Percobaan

NO.	Sensor / Aktuator	Waktu pembacaan / pengeksekuisan (xx diganti 2 digit NIM anda)
1.	Sensor 1	xx00 ms
2.	Sensor 2	1xx0 ms
3.	Sensor 3	2xx ms

FORMAT LAPORAN MODUL 2
NOMER MODUL
NAMA MODUL

NAMA		:
NIM		:
KELAS		:
TANGGAL PRAKT		:
TANGGAL ACC		:

A. Tujuan

B. Landasan Teori

- ○ *Jelaskan tentang arduino dan pemrograman dasarnya*
- ○ *Jelaskan dan carilah literasi datasheet tentang sensor-aktuator yang dipakai pada praktikum*

  Misal : motor servo 9g

  penjelasan misalnya meliputi gambar sensor / aktuator, Pin IO, Dimensi, spesifikasi sensor (lih. datasheet), Buatlah penjelasan secukupnya namun berbobot.
- ○ *Jelaskan dan carilah literasi tentang library-library yang dipakai pada praktikum*

  Misal : Library arduino #include <Servo.h>
- ○ Jelaskan dan carilah literasi tentang RTOS, misal meliputi cara kerja & aplikasinya (lih. Catatan nicolas pantano & klik open source library github misal : github.com/feilipu/Arduino_FreeRTOS_Library atau github.com/greiman/FreeRTOS-Arduino).

C. Hasil Praktikum

C.1 Percobaan 1 blink led & analog read

C.1.1 Skema rangkaian
C.1.2 Foto rangkaian
C.1.3 Script program

C.2 Percobaan 2 RTOS

C.2.1 Skema rangkaian
C.2.2 Foto rangkaian
C.2.3 Script program
C.2.4 Foto hasil (rangkaian saat *running*)

A. Analisa Hasil

D.1 Percobaan 1 blink led & analog read

D.1.1 Analisa script perbaris *(berikan komentar pada sketch code arduino)*
D.1.2 Analisa cara kerja rangkaian

D.2 Percobaan 2 RTOS

D.2.1 Analisa script perbaris *(berikan komentar pada sketch code arduino)*
D.2.2 Analisa cara kerja rangkaian

D.3 Perbandingan free RTOS arduino & pemrograman arduino biasa

Terangkanlah perbedaan, keuanggu, dan kelebihaanya.

B. Kesimpulan

Simpulkan praktikum modul 2 min. 3 paragraf secara komperensif

MINI PROJECT
UTS Praktikum sistem operasi mikroprosesor

Pengambilan nilai UTS SOM
Note : bawa flashdisk berisikan library **freeRTOS Arduino**
Praktikan diberi wakut 20 menit untuk membuat sketch arduino menggunakan **library freeRTOS**
Dengan ketentuan :

Buatlah 5 task pembacaan sensor / aktuator pada arduino dimana waktu pembacaan menjadi parameter penting,
Berdasarkan tabel berikut, buatlah sketch pada IDE arduino :

NO.	Sensor / Aktuator	Waktu pembacaan / pengeksekuisan (x diganti digit NIM anda)
1.	Sensor 1	x000 ms
2.	Sensor 2	1x00 ms
3.	Sensor 3	xxxx ms
4.	Aktuator 1	500x ms
5.	Aktuator 2	300x ms

Perhatikan cara membuat task, meliputi nama, alokasi memory, prioritas serta fungsi task
Tempat : lab komputer TE UMS gedung H sayap kiri

1. Jelaskan tentang pemrograman dasar arduino yang anda ketahui? 5%
2. Jelaskan apa itu interrupt tentang pada arduino! 5%
3. Jelaskan perbedaan RTOS dengan pemrograman biasa pada arduino! 5%
4. Apa yg anda ketahui tentang Jitter? 5%

Link my be used :

- https://arduinomodules.info/ *untuk mengetahui modul sensor-sensor arduino*
- https://playground.arduino.cc/Code/Timer1 *penjelasan tenatan timer & PWM pada arduino*
- https://www.arduino.cc/reference/en/#functions *penjelasan fungsi-fungsi dasar pada IDE arduino*
- http://sensorkit.en.joy-it.net/index.php?title=Main_Page *penjelasan sensor sensor yang ada di lab TE*