|  |  |  |
| --- | --- | --- |
| **KELOMPOK** | 1 | **ACC** |
| **Tanggal Praktikum** | 4 Mei 2018 |  |
|  | Khairunnisa Sekar R. (D400150005) |  |
|  | Dian Arieska (D400150141) | **Tanggal ACC :** |
|  | Nindya Kaloka (D400170019) | **Revisi Tanggal :** |
|  | Saiful Arifin Wahyu W. (D400160065) |  |

**MODUL 4**

Management Resource dalam FreeRTOS

1. **TUJUAN**
2. Praktikan dapat memahami dan mengetahui fungsi Management Resource.
3. Praktikan dapat membuat program dengan fungsi Management Resource.
4. Praktikan dapat membuat program dengan 3 task dengan menggunakan fungsi Management Resource dalam FreeRTOS.
5. **ALAT DAN BAHAN**
6. Laptop / PC yang telah diinstal software Arduino
7. Arduino Uno
8. **DASAR TEORI**

Dalam penggunaan ini, nilai hitungan menunjukkan jumlah sumber daya yang tersedia. Untuk mendapatkan kontrol sumber daya, tugas harus terlebih dahulu mendapatkan semaphore - mengurangi nilai hitungan semaphore. Ketika nilai hitungan mencapai nol tidak ada sumber daya gratis. Ketika tugas selesai dengan sumber daya itu 'memberi' semaphore kembali - incrementing nilai hitungan semaphore. Dalam hal ini diharapkan agar nilai hitung awal sama dengan nilai hitung maksimum, yang menunjukkan bahwa semua sumber daya gratis.

**Perencanaan Sumber Daya**

Sumber daya terbatas, jadi harus merencanakan penggunaannya. Hanya dapat menempatkan setiap orang di satu tempat, pada satu waktu, mengerjakan satu tugas, sesuai dengan keahliannya yang unik. Perencanaan sumber daya yang efektif membantu memberikan proyek tepat waktu tanpa mengubah tenggat waktu, mengubah prioritas, dan evolusi proyek umum. Perencanaan sumber daya menjaga tingkat pemanfaatan pada tingkat yang menguntungkan, mengurangi waktu yang tidak dapat ditagih yang dihabiskan oleh sumber daya untuk proyek.

**Keuntungan Menggunakan Mnaagement Resource**

Perangkat lunak perencanaan sumber daya memecahkan tantangan khusus untuk mengetahui sesuatu ditempatkan di tempat yang tepat untuk memenuhi kebutuhan klien dan mencapai tujuan bisnis. Ini membantu mengukur pemanfaatan, sehingga apakah cukup, terlalu banyak, atau terlalu sedikit. Dapat memastikan bahwa mereka menyelesaikan pekerjaan itu tepat waktu, dan dapat mengalokasikannya kembali ketika jadwal proyek bergeser. Selain itu, perangkat lunak perencanaan sumber daya membantu memperkirakan skala. Ketika sebuah proyek berlipat ganda dalam ukuran atau memenangkan klien baru, perangkat lunak perencanaan sumber daya menunjukkan di mana dapat menarik sumber daya untuk mencapai tugas-tugas yang diperlukan. Beberapa aplikasi perencanaan sumber daya menggabungkan informasi keuangan atau berintegrasi dengan solusi CRM dan akuntansi. Ini memungkinka melihat manfaat-biaya dari pengalokasian ulang sumber daya untuk tugas-tugas lain, serta kesehatan keuangan keseluruhan. Akhirnya, perangkat lunak perencanaan sumber daya menawarkan ruang kerja pusat berkolaborasi.

**Kerugian Menggunakan Management Resource**

Salah satu hambatan terbesar untuk mengadopsi perangkat lunak perencanaan sumber daya khusus adalah integrasi mahal yang diperlukan untuk mendapatkan manfaat yang diantisipasi. Dapat mengurangi resiko ini dengan memilih perangkat lunak perencanaan sumber daya yang merupakan bagian dari rangkaian aplikasi, dan / atau yang terintegrasi dengan perangkat lunak manajemen akuntansi dan proyek pilihan di luar rak. Menggunakan perangkat lunak perencanaan sumber daya daripada spreadsheet juga akan membutuhkan pelatihan (dan kemungkinan rekayasa ulang proses bisnis) untuk berhasil. Pengalaman pengguna bervariasi secara drastis di antara solusi yang berbeda, dan sangat penting untuk memilih satu dengan antarmuka pengguna yang akan digunakan. Karena alasan ini, penting untuk memilih penyedia perangkat lunak yang menunjukkan antarmuka pengguna dan penawaran layanan pelanggan yang memadai yang memenuhi standar.

1. **HASIL PRAKTIKUM**

D.1 Hasil Percobaan Script

#include "FreeRTOS\_AVR.h"

static void prvPrintTask (void \*pvParameters);

static void prvNewPrintString(const char \*pcString);

SemaphoreHandle\_t xMutex;

void setup() {

Serial.begin(9600);

xMutex = xSemaphoreCreateMutex();

srand(100);

if(xMutex !=NULL);

{

xTaskCreate(prvPrintTask, "Print 1", 200, (void\*)"Task 1 Running \r\n",1,NULL);

xTaskCreate(prvPrintTask, "Print 1", 200, (void\*)"Task 2 Running \r\n",2,NULL);

xTaskCreate(prvPrintTask, "Print 1", 200, (void\*)"Task 3 Running \r\n",3,NULL);

vTaskStartScheduler();

}

for(;;);

}

static void prvNewPrintString(const char \*pcString)

{

xSemaphoreTake(xMutex,portMAX\_DELAY);

{

Serial.print(pcString);

Serial.flush();

}

xSemaphoreGive(xMutex);

if(Serial.available())

{

vTaskEndScheduler();

}

}

static void prvPrintTask(void \*pvParameters)

{

char \*pcStringToPrint;

pcStringToPrint = (char \*) pvParameters;

{

prvNewPrintString (pcStringToPrint);

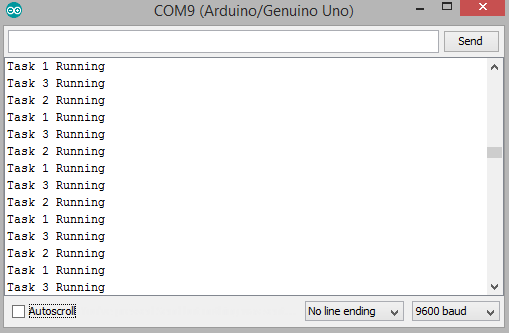
vTaskDelay((rand()& 0x1FF));

}}

void loop (){

}

D.2 Hasil Tampilan Serial Monitor



1. **ANALISA**

E.1. Analisa Percobaan

Pada praktikum 4 ini menggunakan fungsi management resource dan 3 task pada program. Program dibuat dapat berjalan secara terus menerus secara gantian pada masing- masing task nya. xSemaphoreGive digunakan untuk memasukkan data kemudian akan diambil oleh xSemaphoreTake dan kemudian ditampilkan pada Serial Monitor. Sehingga program berjalan secara terus menerus (forever loop) pada serial monitor.

vTaskDelay digunakan untuk menampilkan data dengan penundaan pemunculan data di serial monitor. Pada setiap task program tidak menggunakan delay pada xSemaphoreGive dan XsemaphorTake karena agar dapat berjalan saling bergantian. Pada Serial.println digunakan untuk menapilkan data keserial monitor sesuai urutan yang ditentukan sesuai prioritas. Sedangkan untuk fungsi dari xTaskCreate(prvPrintTask, "Print 1", 200, (void\*)"Task 1 Running \r\n",1,NULL); digunakan untuk meng*create* Task dan menentukan nomor prioritas untuk pemunculan pada serial monitor.

SemaphoreHandle\_t xMutex digunakan untuk mengendalikan management resource yang kita buat. Sehingga program berjalan secara terus menerus (forever loop) pada serial monitor secara bergantian tanpa ada yang keluara yang double.

1. **KESIMPULAN**
2. Semaphore adalah suatu isyarat yang digunakan untuk menentukan akses untuk berbagi sumber daya sistem.
3. xSemaphoreGive dan xSemaphoreTake dalam praktikum ini saling berhubungan dalam jalannya program
4. Dari Serial Monitor terlihat hasil dari xSemaphoreGive dan xSemaphoreTake secara teratur karena adanda pengaturan management resource pada SempahoreHandle\_t xMutex.