|  |  |  |
| --- | --- | --- |
| **KELOMPOK** | **E/5** | **ACC** |
| **Tanggal Praktikum** |  | **Asisten 1** |
| **Asisten 2** |
|  | 1. **IBNU S K (D400150093)** |  |
|  | 1. **TAUFIQ D (D400150102)** | **Tanggal ACC :** |
|  | 1. **AGUS YOGA (D400150135)** | **Revisi Tanggal :** |

**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 digunakan. Untuk mendapatkan kontrol sumber daya, aplikas harus terlebih dahulu mendapatkan semaphore - mengurangi nilai hitungan semaphore. Ketika nilai hitungan mencapai nol tidak ada sumber daya yang digunakan. 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 dapat digunakan.

**Perencanaan Sumber Daya**

Sumber daya pada Arduino terbatas seperti clock atau perintah delay. Sehingga dibutuhkan perencanaan sumber daya agar dapat menjalankan aplikasi dalam waktu yang ditentukan tanpa adanya pergeseran waktu diluar ketentuan. Penggunaan aplikasi yang banyak tentu akan mengurangi performa board maka dibutuhkan perencanaan sumber daya yang baik.

**Keuntungan Menggunakan Management Resource**

Perencanaan sumber daya dapat menyelesaikan masalah yaitu mengetahui apakah aplikasi dapat bekerja sesuai kebutuhan. Dengan perencanaan sumber daya dapat mengetahui apakah waktu yang digunkan cukup atau tidak pada saat aplikasi berjalan. Selain itu, dapat memastikan bahwa aplikasi berjalan dengan waktu yang tepat dengan melihat hasilnya pada saat aplikasi berjalan. Skala prioritas yang sematkan pada setiap aplikasi akan berjalan sesuai dengan nilai prioritas yang diberikan sehingga dapat mengatur kapan waktu setiap aplikasi berjalan meski aplikasi yang berjalan lebih dari satu.

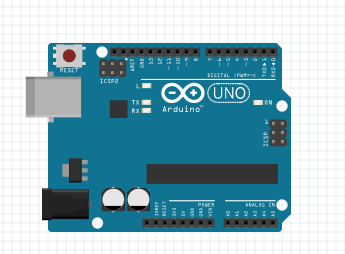
**Kerugian Menggunakan Management Resource**

Memerlukan keahlian lebih untuk dapat membuat perintah pada board karena menggunakan peritah perintah yang tidak biasa digunakan pada board. Diperlukan ketelitian dalam menganalisa perintah dan melihat setiap aplikasi yang berjalan.

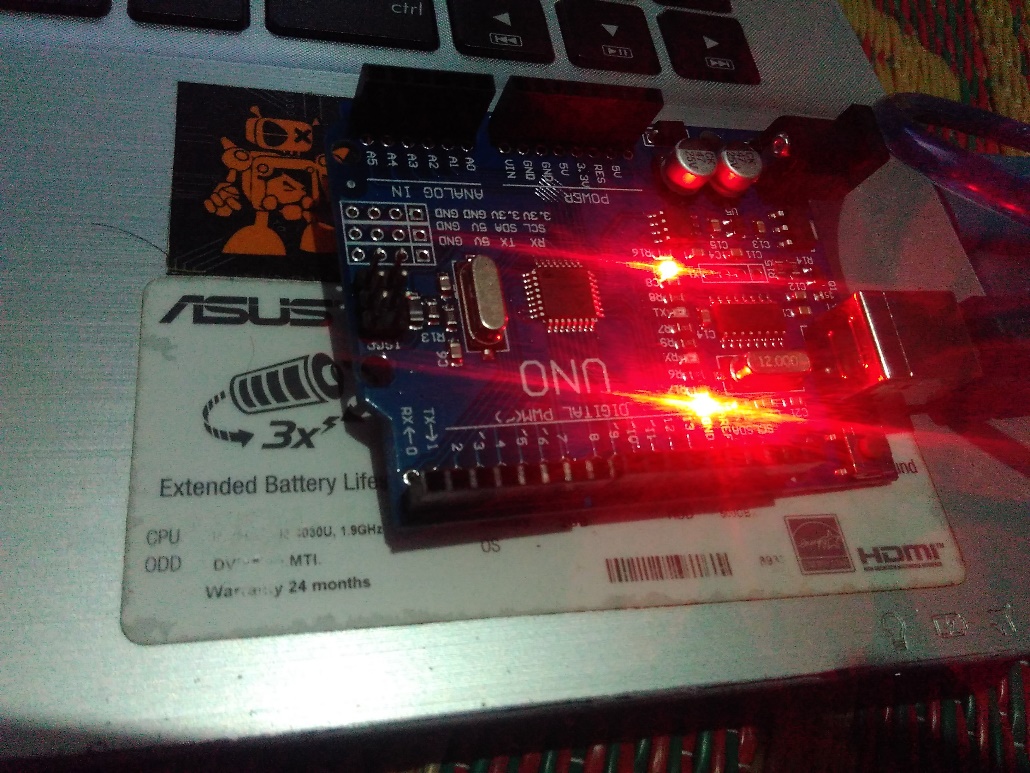
1. **HASIL PRAKTIKUM**

D.1 Hasil Percobaan

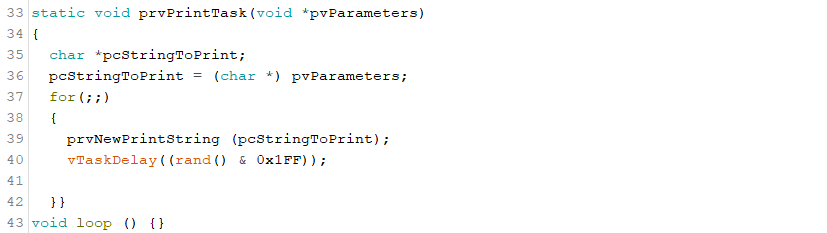
D.1.1 SKEMA RANGKAIAN

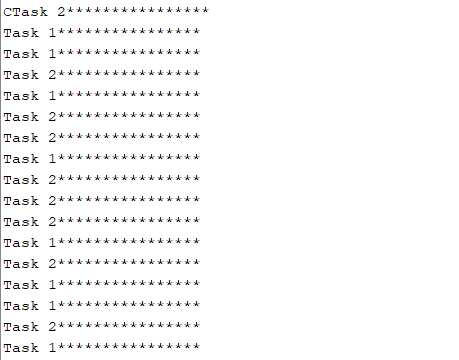


D.1.2 FOTO RANGKAIAN



 D.1.3.SCRIPT PROGRAM



D.1.4. SERIAL MONITOR

1. **ANALISA**

Praktikum ini menerapkan fungsi management resource atau perencanaan sumber daya dan menggunakan 3 task. Program dibuat dapat berjalan secara terus menerus secara gantian pada masing- masing tasknya.

xSemaphoreGive digunakan untuk memasukkan atau memberikan data pada task, kemudian data akan diambil dengan xSemaphoreTake

vTaskDelay digunakan untuk menampilkan data dengan penundaan pengambilan data. Pada setiap task program tidak menggunakan delay pada xSemaphoreGive dan XsemaphorTake karena agar dapat berjalan saling bergantian.

Serial.println digunakan untuk menapilkan data keserial monitor sesuai urutan yang ditentukan sesuai prioritas.

xTaskCreate(prvPrintTask, "Print 1", 200, (void\*)"Task 1 Running \r\n",1,NULL); digunakan untuk meng*create* Task dan menentukan nomor prioritas data digunakan.

SemaphoreHandle\_t xMutex digunakan untuk mengendalikan management resource yang dibuat.

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