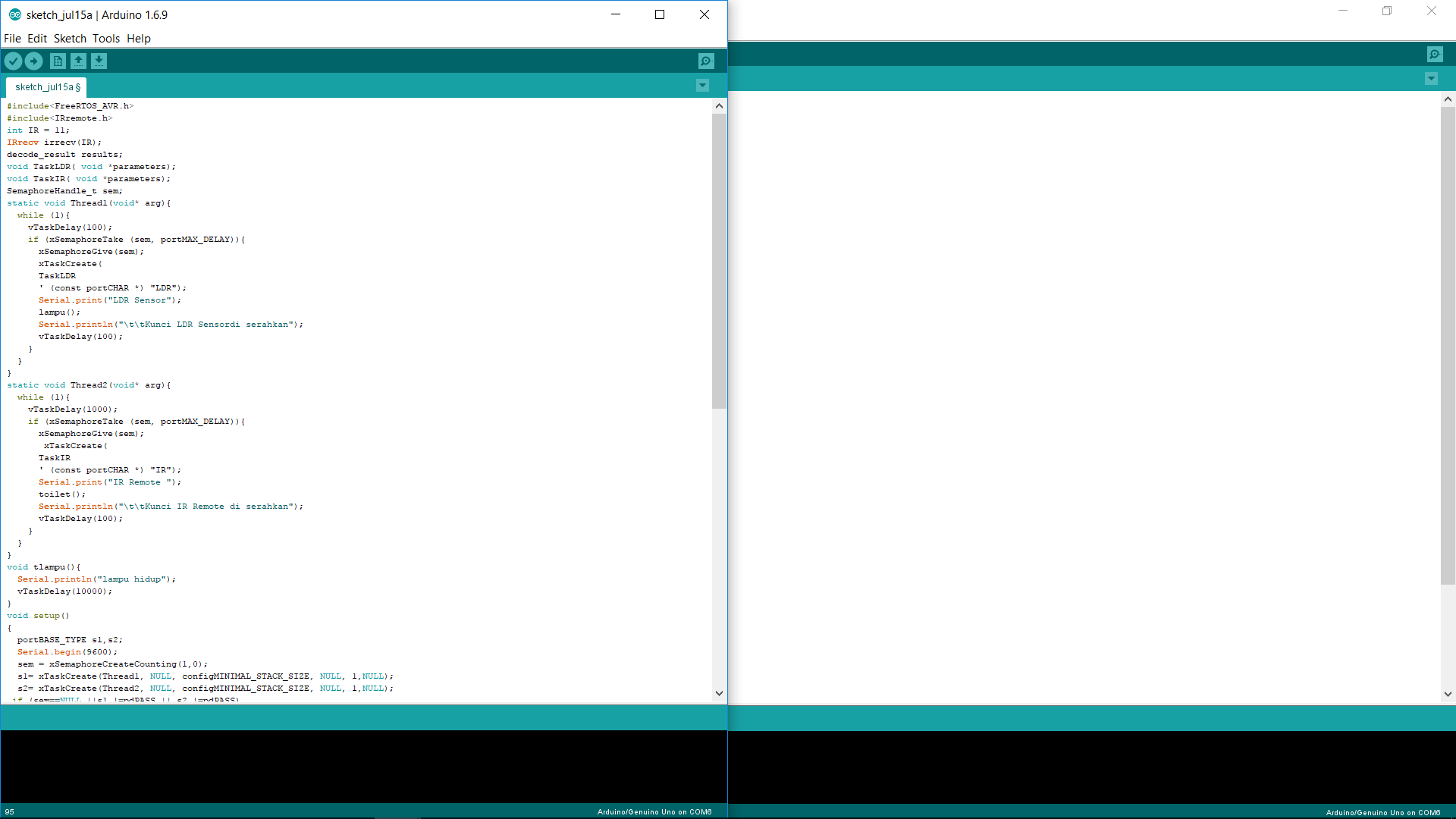
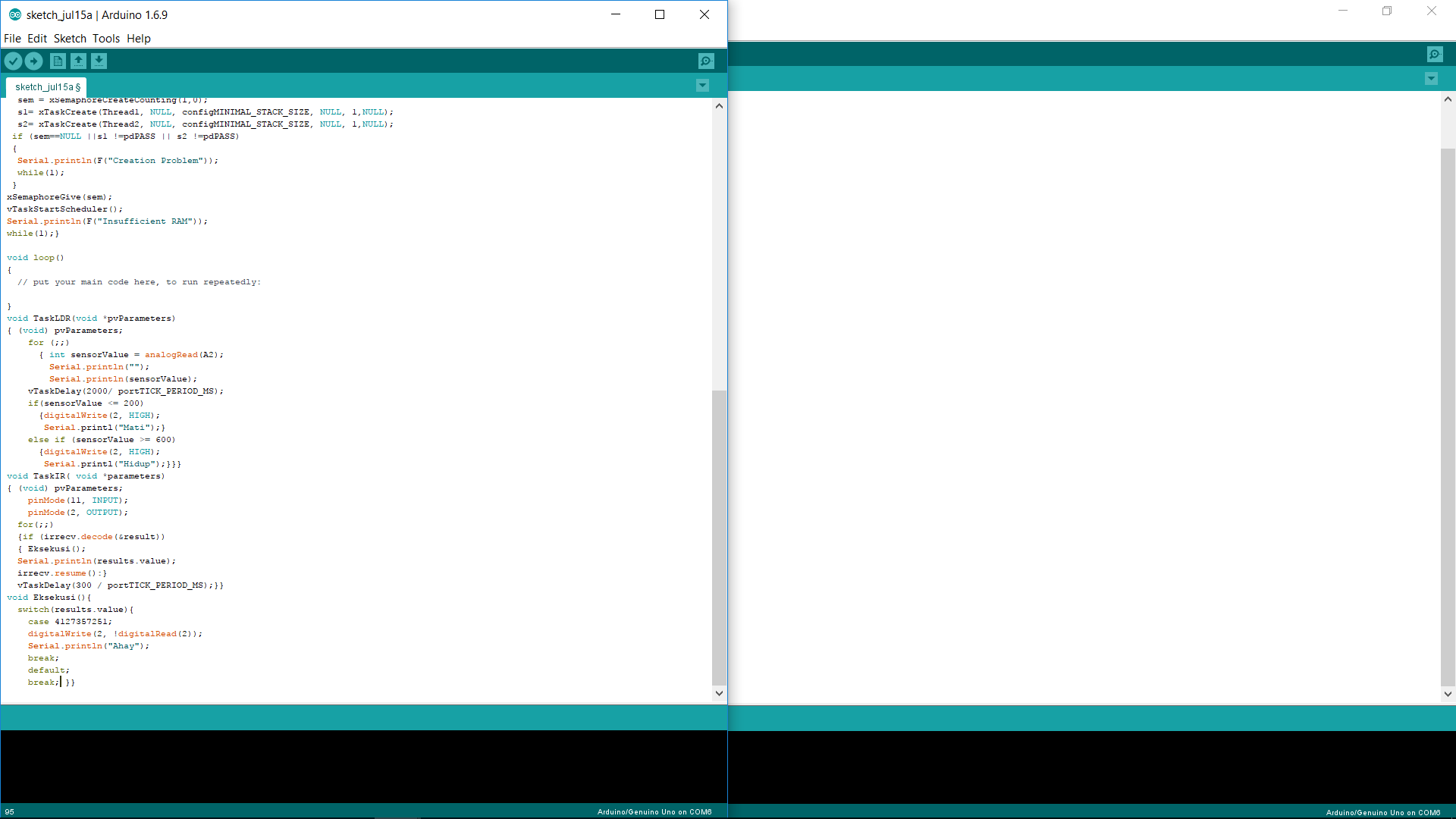
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Group |  |  |  | ACC |
| Date of Lab Work |  |  |  |  |
| Name | 1 | Pratama Aji Nur Rahman | D400154003 |
|  | 2 | Amnaduny Akhara | D400154006 |
|  | 3 | Milzam Wafi Azhar | D400154007 | ACC Date : |
|  | 4 | Jeski Saputra | D400154009 | Revise Date : |

MODULE 2

SEMAPHORE

1. PURPOSE
2. Which FreeRTOS API functions can be used from within an interrupt service routine.
3. Methods of deferring interrupt processing to a task.
4. How to create and use binary semaphores and counting semaphores.
5. The differences between binary and counting semaphores.
6. SCRIPT





1. ANALYSIS

On the result of semaphore program is given command 2 condition and each condition will work if get key, when thread 1 exit from toilet, thread 1 Must give Key to thread 2 with the result thread 2 can enter toilet.

1. CONCLUSION
2. A task is a software feature that is unrelated to the hardware on which FreeRTOS is running. The priority of a task is assigned in software by the application writer, and a software algorithm (the scheduler) decides which task will be in the Running state.
3. Although written in software, an interrupt service routine is a hardware feature because the hardware controls which interrupt service routine will run, and when it will run. Tasks will only run when there are no ISRs running, so the lowest priority interrupt will interrupt the highest priority task, and there is no way for a task to pre-empt an ISR.