## In Lab Task 1

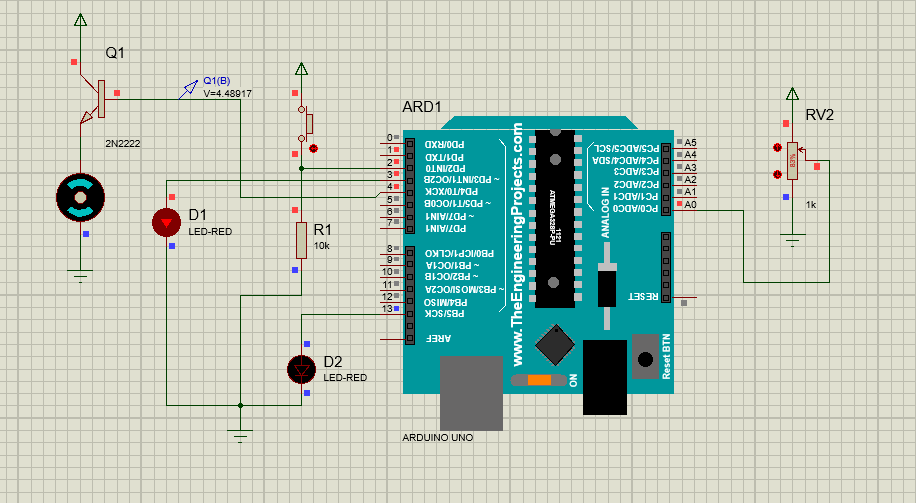
**Please see and analyze the above task and complete it using RMS using freeRTOS library in Arduino with following specs controlled simultaneously by various tasks.**

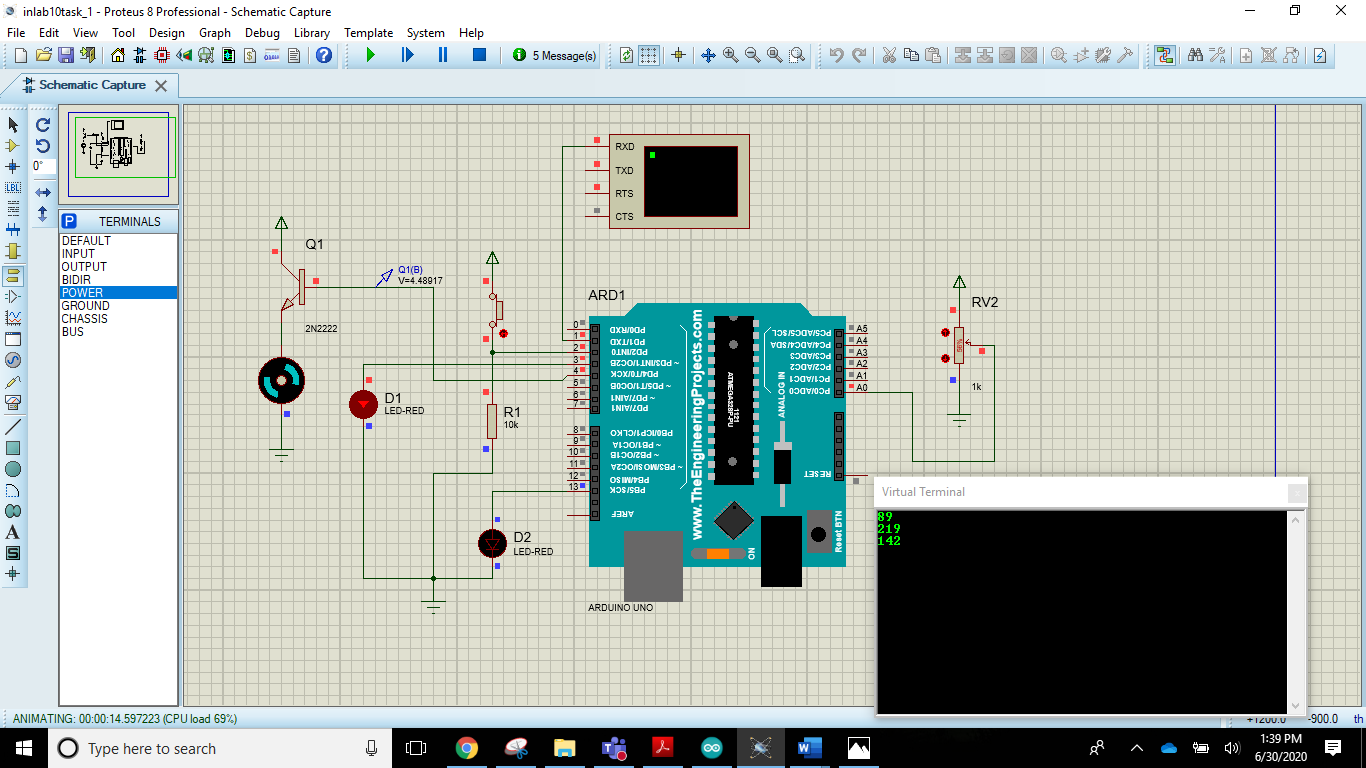
**1) Variable resistor is controlling the speed of the DC Motor.**

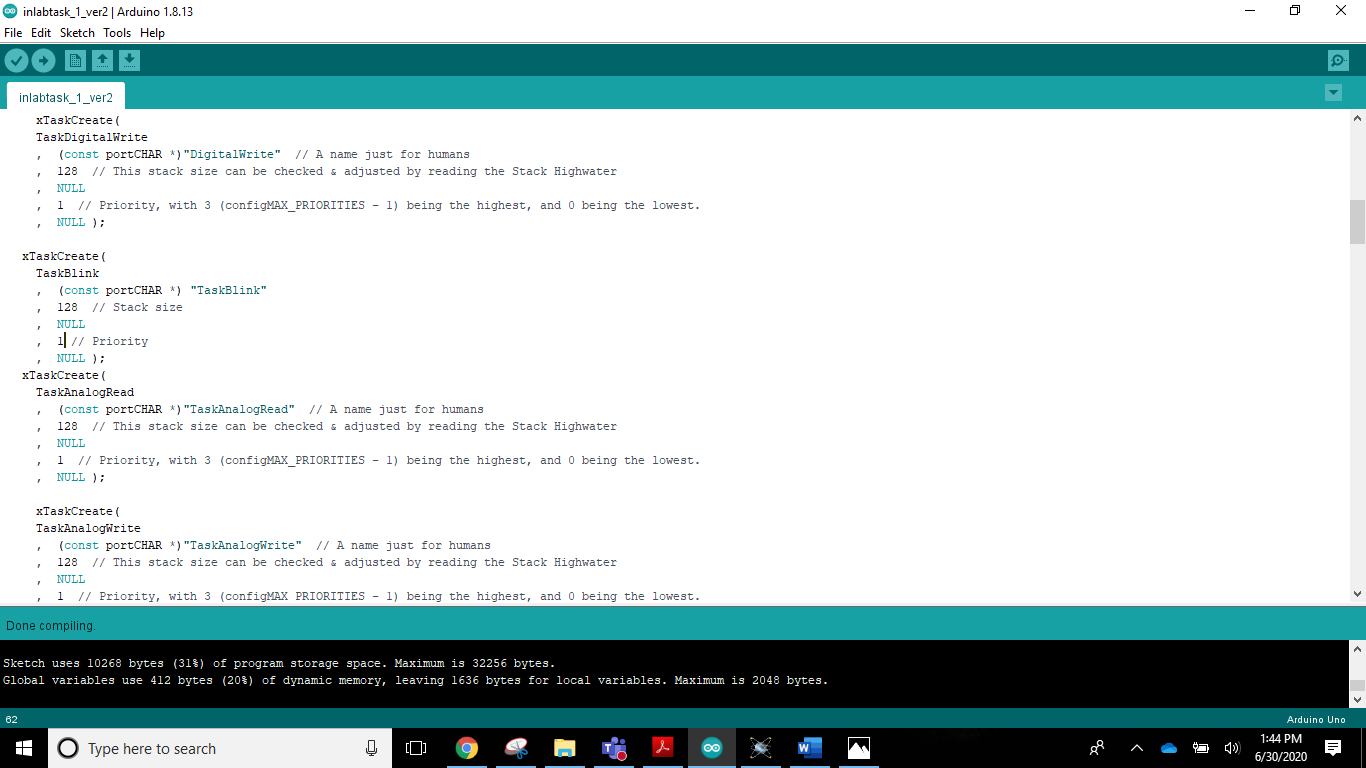
**2) Push button is controlling the LED turn ON or OFF.**

**3) And one LED is consistently blinking after two seconds.**

**Using 1 semaphore and Round Robin (Equal Priorities):**



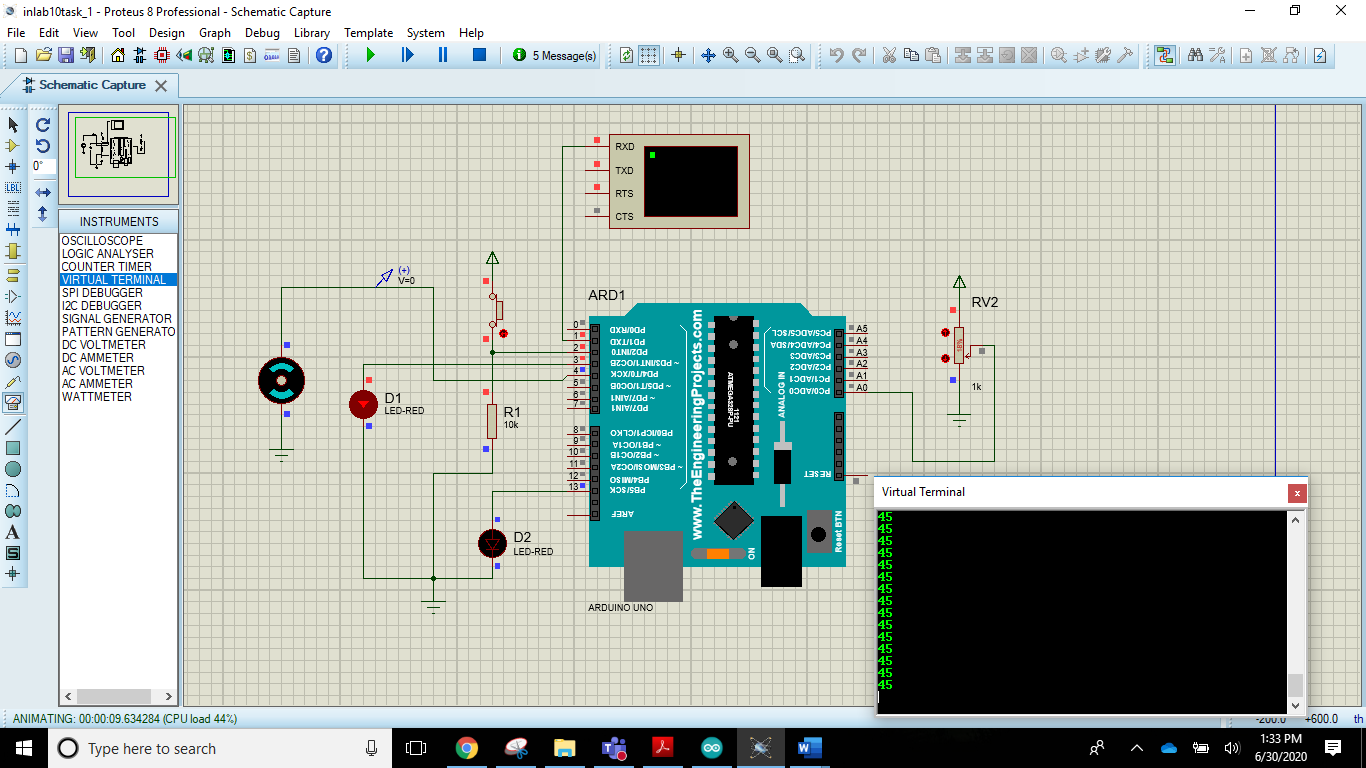


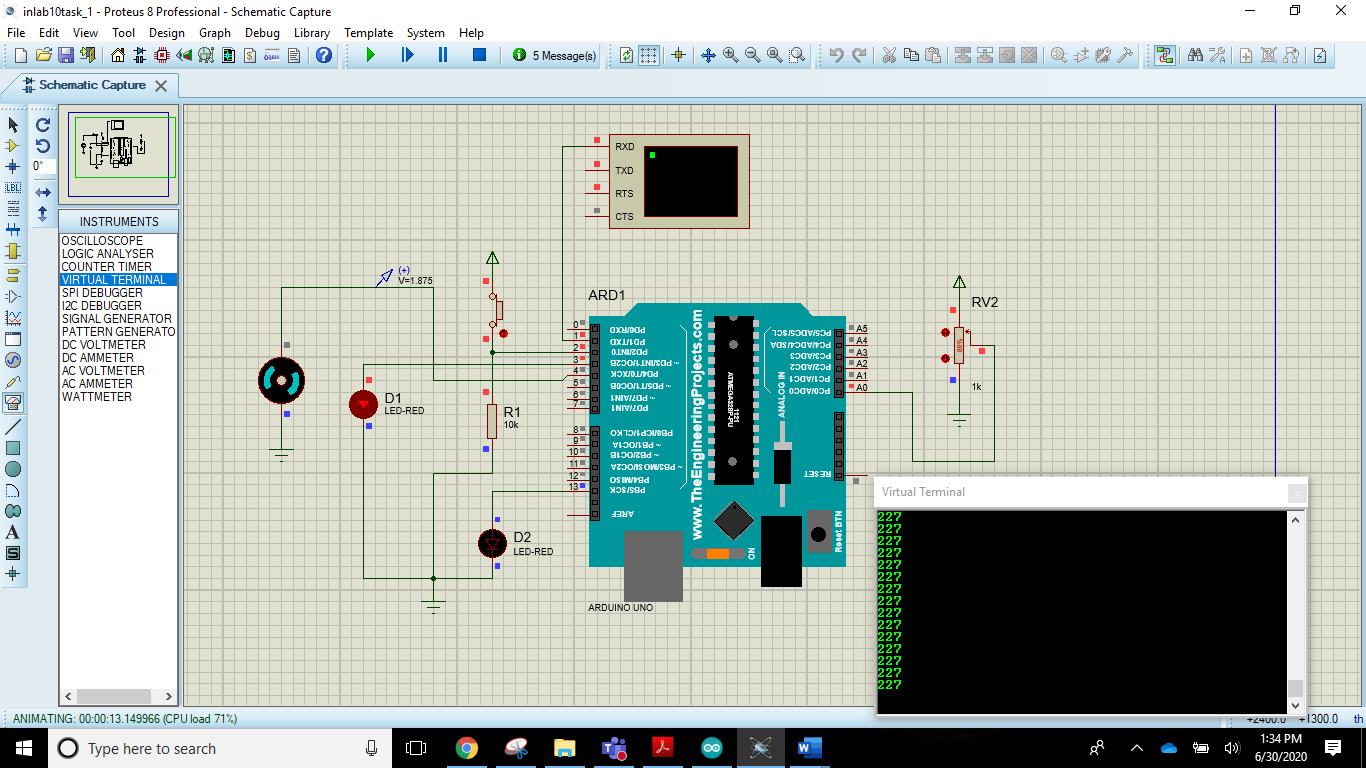


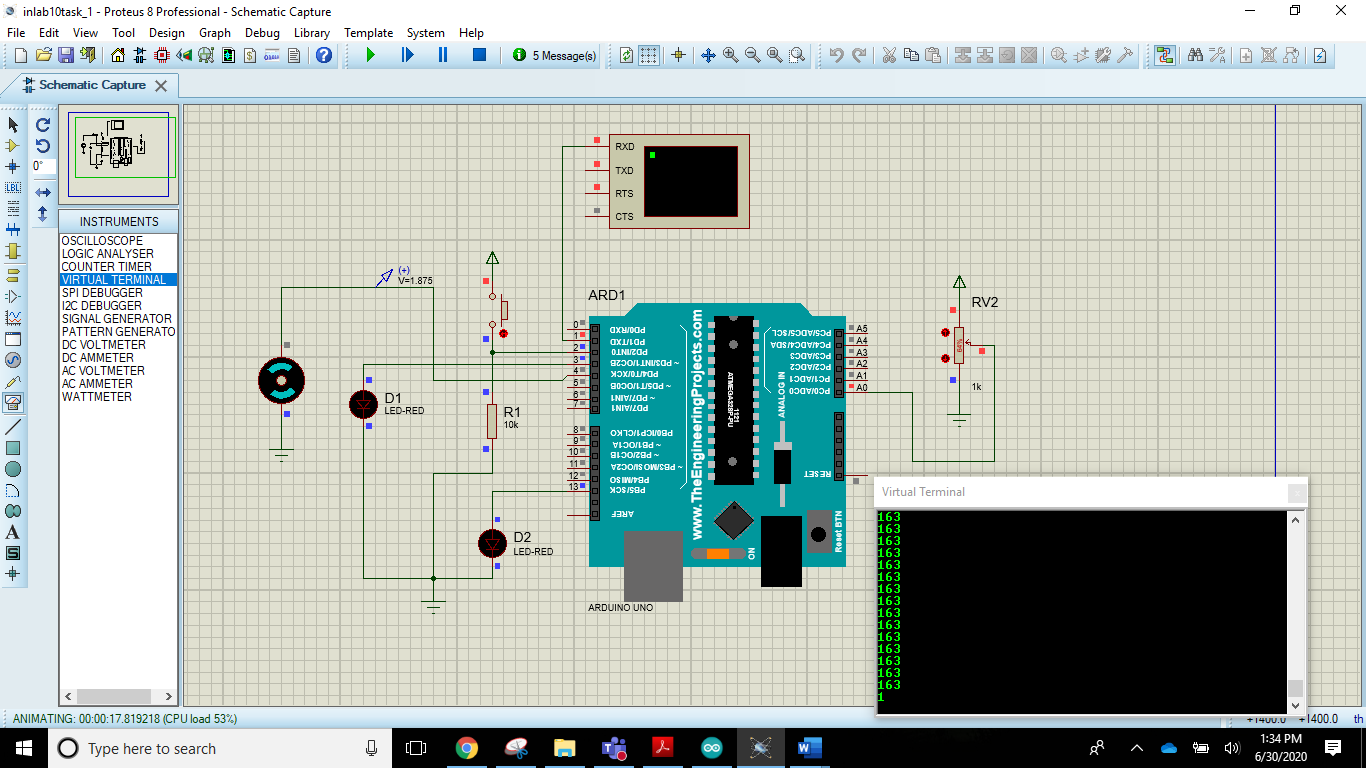
**Analysis:**

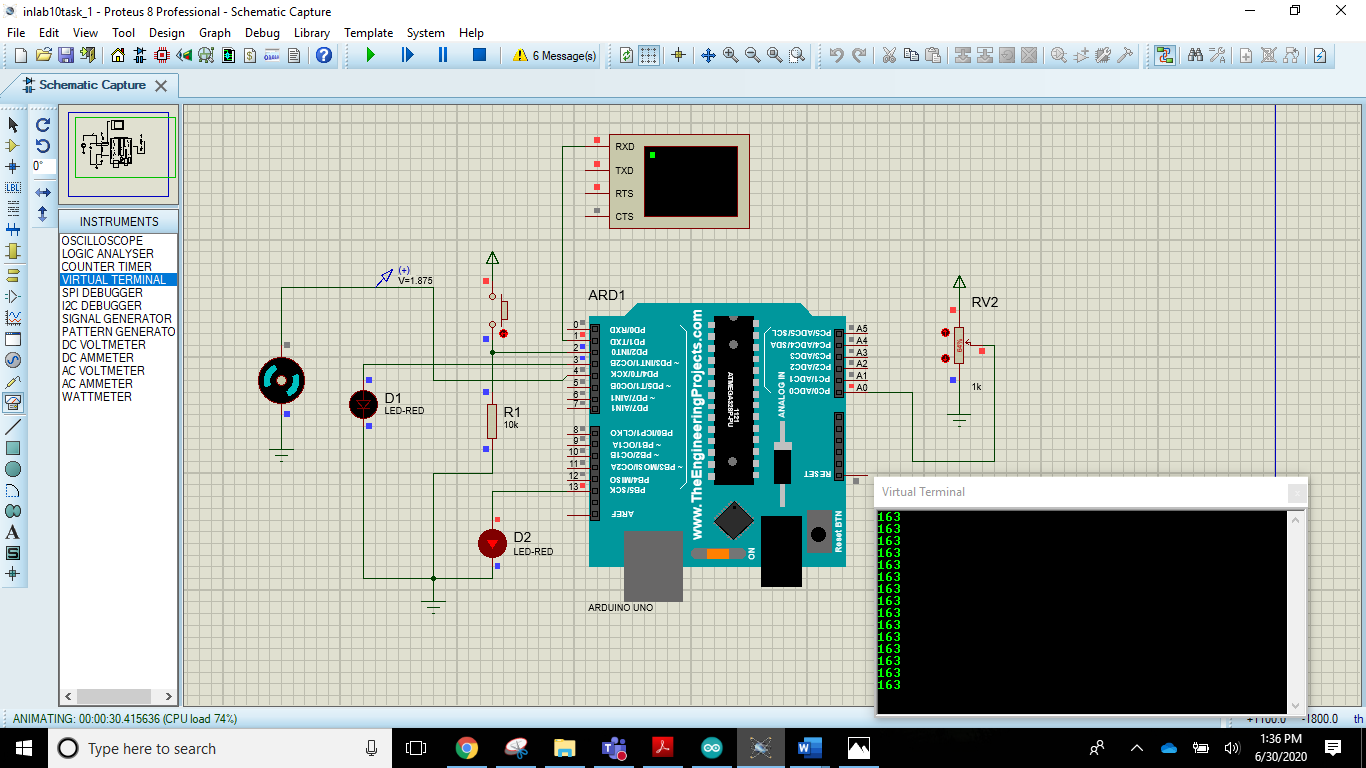
|  |
| --- |
| Speed of motor does changes but very slowly as we see at the virtual terminal we vary the pot value. All the other tasks are running but with a lot of delay.  The system Is overloaded and the scheduling is poor. |

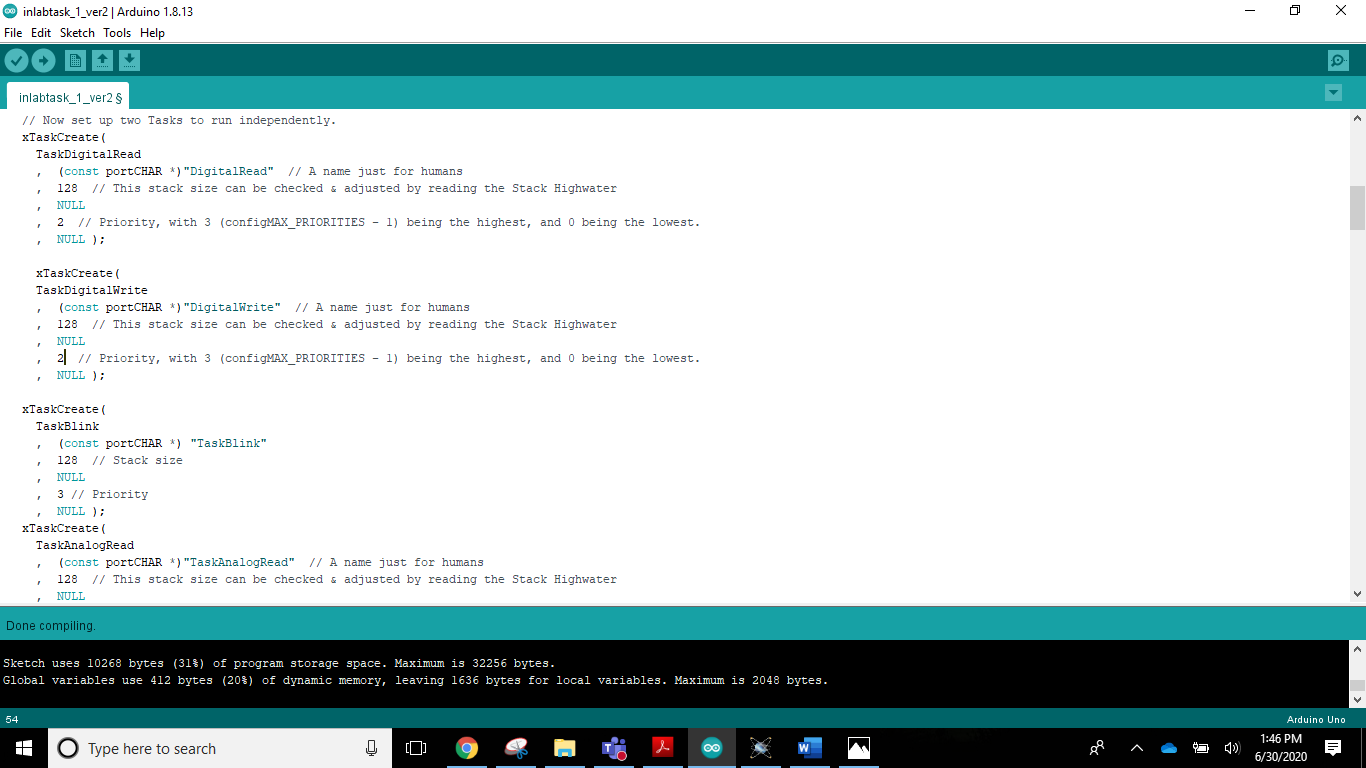
**Using 1 semaphore and RMS Scheduling:**

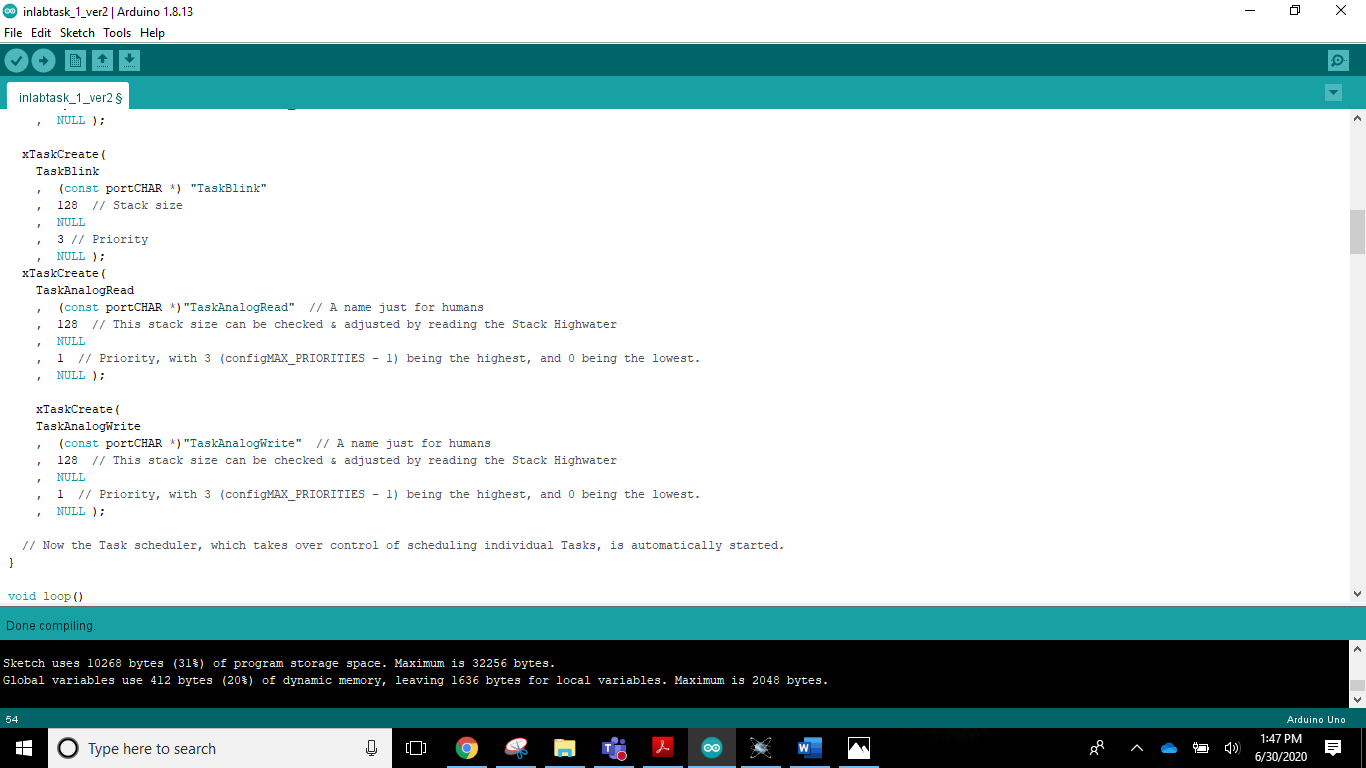












**Analysis:**

|  |
| --- |
| Speed of motor changes as soon as we change the potentiometer value, all the other tasks are also running without any delay.  All the tasks are running smoothly. |

## Conclusion:

|  |
| --- |
| When we have a lot of tasks to be scheduled, it is better to use RMS scheduling than Round Robin Scheduling. This makes efficient use of resources and tasks can be completed within their time period without any delay in execution time.  In Round Robin technique, if we have a lot of tasks, the scheduler/kernel takes up a lot of delay after each task to decide which task should be executed next. The clock interrupt is too much which causes poor system performance. |