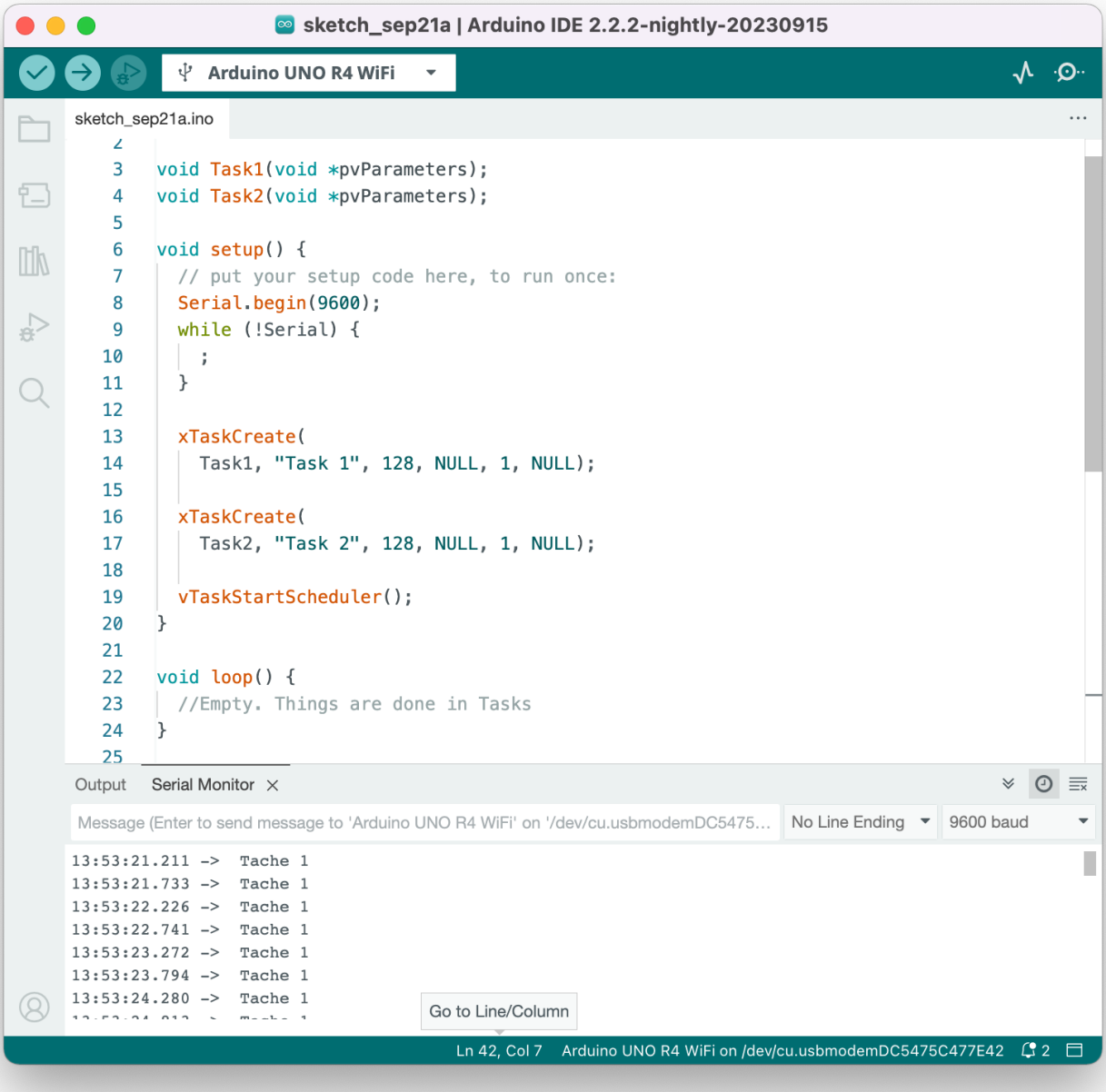
Thomas PEUGNET – Vincent LAGOGUÉ – David TEJEDA
M1 Réseaux et Sécurité – RS3

Programmation multi-tâches – Rendu TP02

Partie 1

Le code ci-dessous nous permet d'obtenir le résultat affiché dans la console.

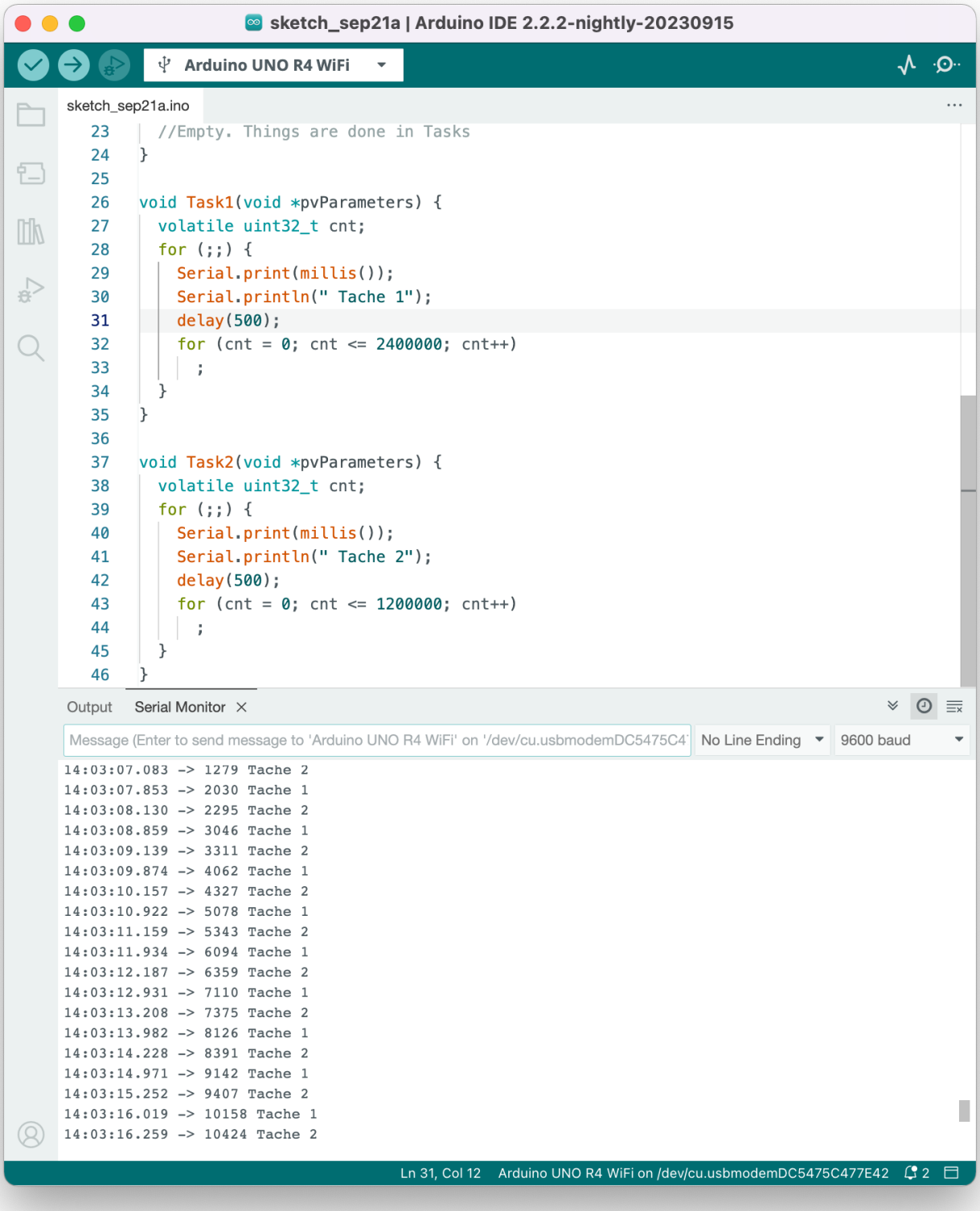


Avec Task1 et Task2 définies plus bas dans le code. Nous pouvons noter que le temps d'exécution est variable, mais dépasse rarement les 500ms, et descend rarement au-dessous de 20ms. Nous pouvons par ailleurs noter que seule la tâche 1 est visible à l'exécution du programme.

Note : La bibliothèque sur macOS contenant les fonctions de freeRTOS se trouve sur le chemin suivant :

```
/Users/<user>/Library/Arduino15/packages/arduino/hardware/renesas_uno/1.0.4/libraries/Arduino_FreeRTOS/src/Arduino_FreeRTOS.h
```

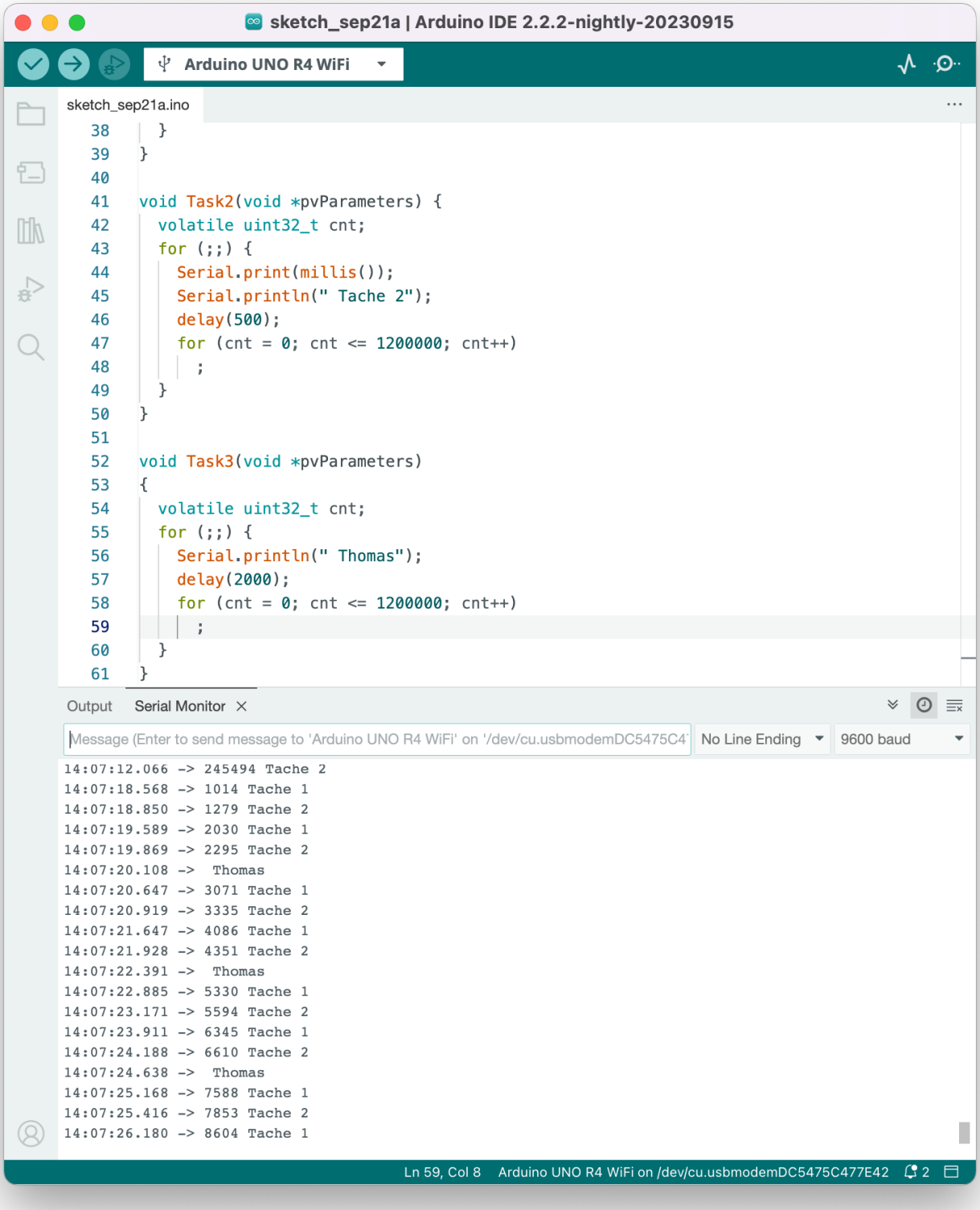
En décommentant les lignes permettant de faire un delay, nous pouvons constater que nous avons bien une « simultanéité » lors de l'exécution des tâches.



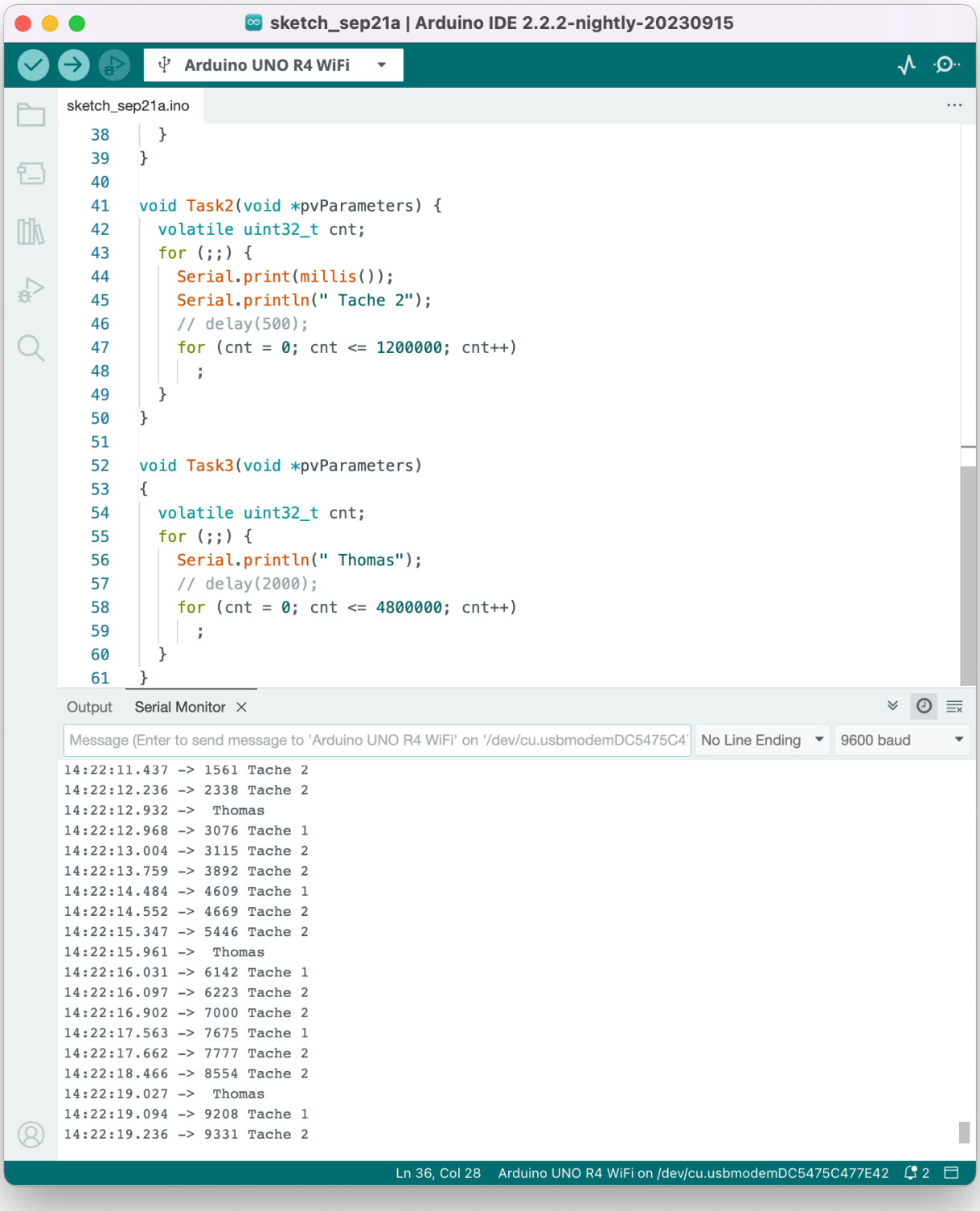
Après avoir commis l'erreur de décommenter le `delay()`, il sera re commenté pour la majorité des exercices suivants.
Par ailleurs, nous pouvons constater les différences entre les vitesses d'exécution entre nos différentes machines. Pour ma part, sur une machine avec un i7 (7eGen), `3 200 000 ~= 2s`.

Exercice 01

La troisième tâche est créée de façon identique à celles précédemment créées précédemment. Il était cependant nécessaire de modifier le délai afin de ne la faire apparaître

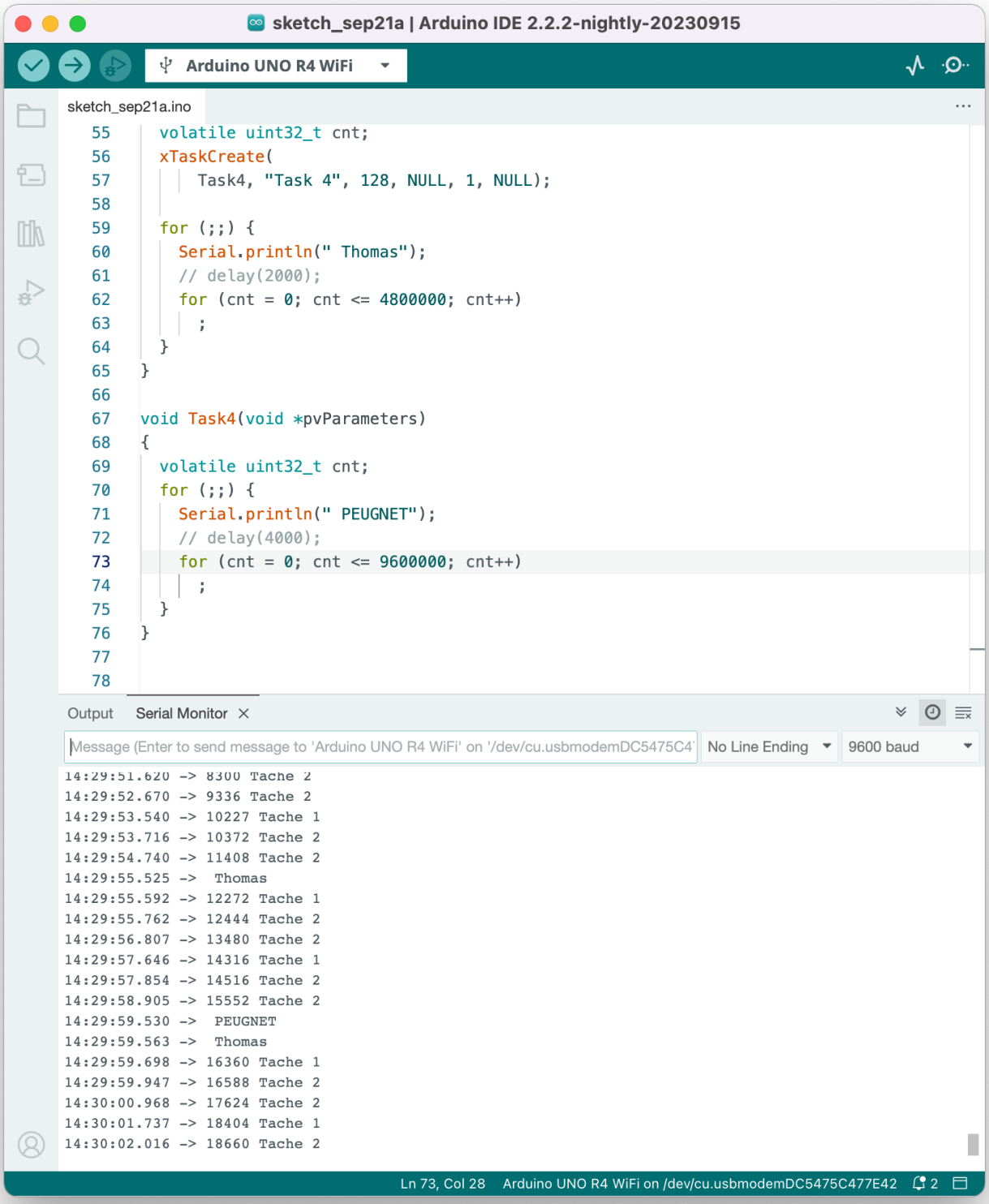


Note : Il n'était pas autorisé d'utiliser delay(), le code modifié est donc le suivant, avec le résultat en console.



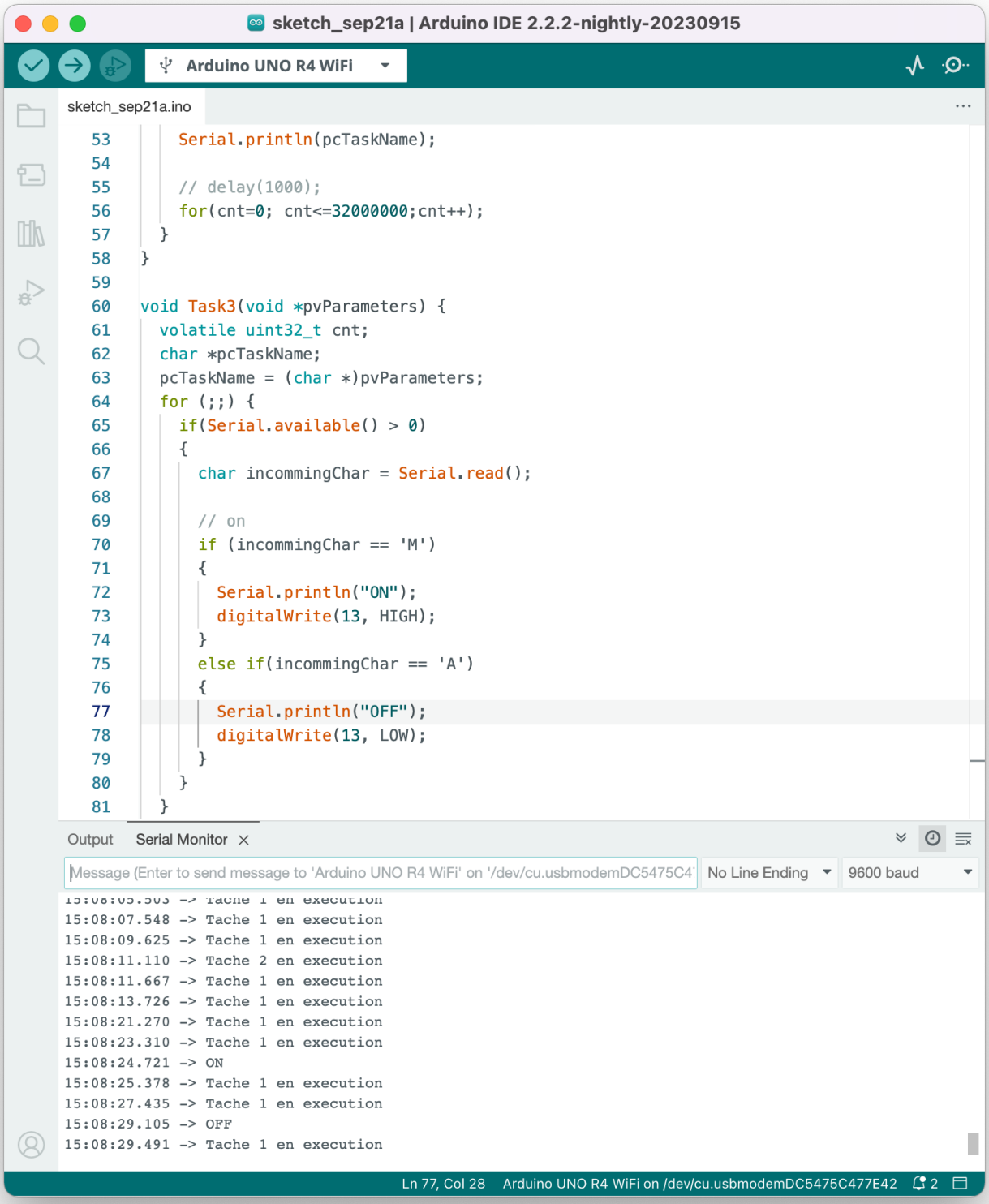
Exercice 02

Le code permettant d'afficher le nom toutes les 4 secondes est le suivant.



Exercice 03

Le code permettant d'expérimenter le programme de l'exercice 07 du TP01 est le suivant.




```
53      Serial.println(pcTaskName);
54
55      // delay(1000);
56      for(cnt=0; cnt<=32000000;cnt++);
57    }
58  }
59
60  void Task3(void *pvParameters) {
61    volatile uint32_t cnt;
62    char *pcTaskName;
63    pcTaskName = (char *)pvParameters;
64    for (;;) {
65      if(Serial.available() > 0)
66      {
67        char incommingChar = Serial.read();
68
69        // on
70        if (incommingChar == 'M')
71        {
72          Serial.println("ON");
73          digitalWrite(13, HIGH);
74        }
75        else if(incommingChar == 'A')
76        {
77          Serial.println("OFF");
78          digitalWrite(13, LOW);
79        }
80      }
81    }
```

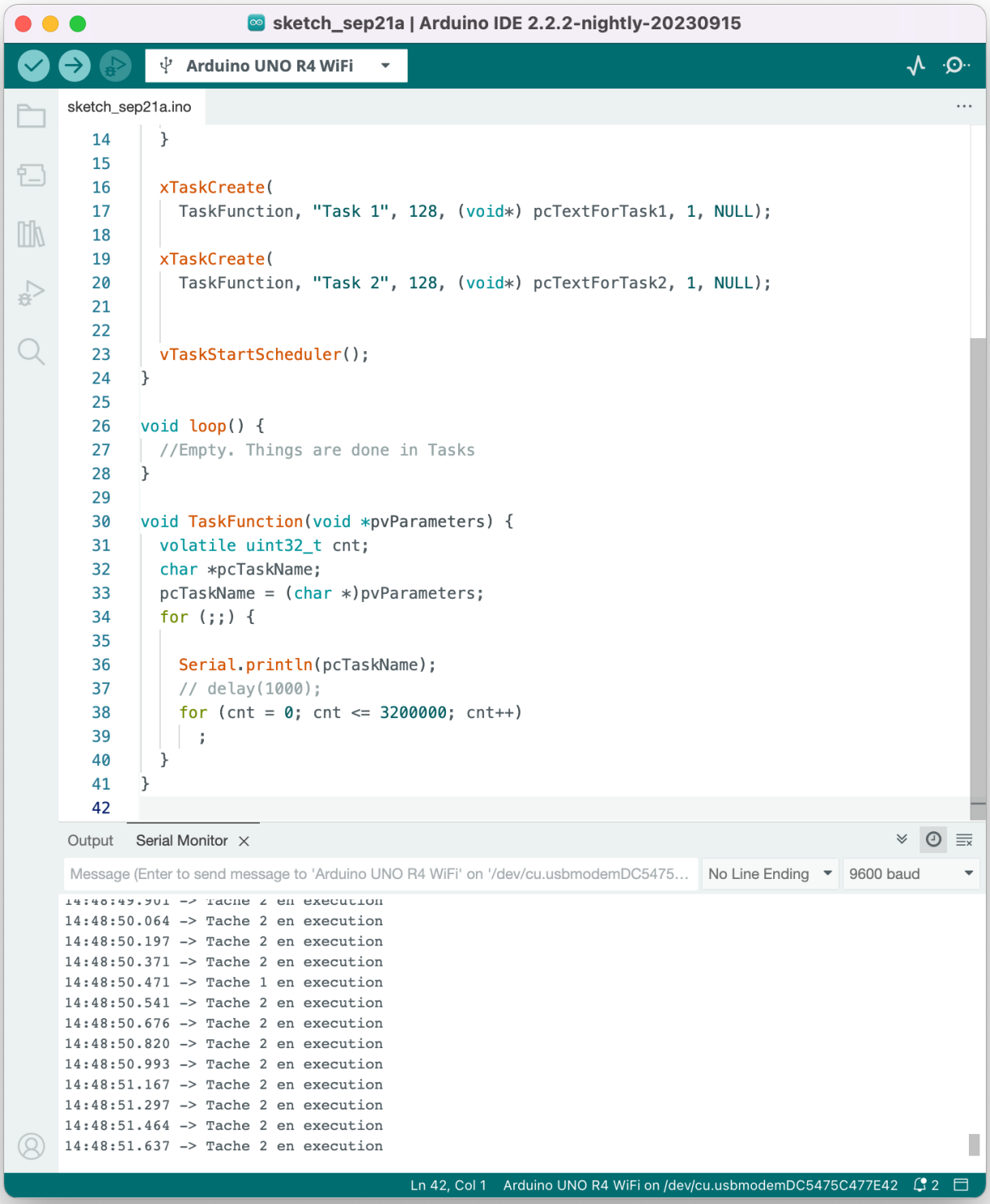
Afin de ne pas devoir faire de photo, nous avons affiché `ON` ou `OFF` lorsque la LED apparaît allumée ou éteinte sur la carte Arduino.

Exercice 05

Le code permettant d'afficher ces deux tâches est le suivant.

Exercice 05 - suite

Le code permettant de refaire l'exercice 05 mais en effectuant une optimisation du code est le suivant.



Nous pouvons constater un comportement identique.

Exemple

Le code suivant permet d'expérimenter le comportement qui aura lieu si jamais la priorité de deux tâches identiques n'est pas la même. Nous pouvons constater que seule la tâche de priorité supérieure est traitée.