PROSIT ALLER

PROJET – SYSTÈMES EMBARQUÉS

PROSIT 2 – UN REGISTRE DE PERFORMANCE

Table des matières

[***1.*** ***Analyse du contexte*** 2](#_Toc114670168)

[***2.*** ***Mots clés*** 2](#_Toc114670169)

[***3.*** ***Problématique*** 2](#_Toc114670170)

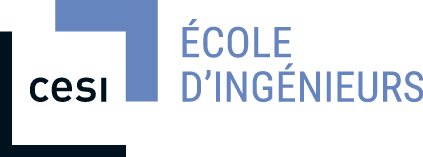
[***4.*** ***Contraintes*** 2](#_Toc114670171)

[***5.*** ***Livrables*** 2](#_Toc114670172)

[***6.*** ***Généralisation*** 3](#_Toc114670173)

[***7.*** ***Pistes de solutions*** 3](#_Toc114670174)

[***8.*** ***Plan d’action*** 3](#_Toc114670175)



# ***Analyse du contexte***

Une société THS propose à ses équipes de développer un projet sur un microcontrôleur. Pour cela, il doit-être donner des instructions au système en utilisant l’assembleur

# ***Mots clés***

* ASM
* CPU
* Microcontrôleur
* Microprocesseur
* Suite de Fibonacci
* Architecture Von Neumann
* Arduino
* AVR
* Horloge RTC
* Moniteur série
* Cycle
* Architecture Harvard
* Transistor
* Porte logique
* Registre
* « Move » de données
* « Add » de données
* Itération
* Unité logique

# ***Problématique***

Comment obtenir le résultat final tout en ayant un programme optimisé ?

# ***Contraintes***

* Avoir le bon résultat : 377
* Utiliser ASM
* Respecter l’architecture du schéma
* PNG (Fibonacci)
* Pas de binaire

# ***Livrables***

Code complet

# ***Généralisation***

Apprendre le fonctionnement d’un microcontrôleur

# ***Pistes de solutions***

* Langage assembleur
* S’appuyer sur la suite de Fibonacci simplifiée/addition des deux derniers nombres
* Regarder si la librairie ASM est native ou non
* L’erreur provient peut-être du nombre d’itérations

# ***Plan d’action***

* Copier le programme
* Comprendre le programme avec des « print »
* Comprendre l’architecture du schéma
* Comprendre et utiliser l’assembleur
* Compléter le programme
* Comprendre et corriger l’erreur du programme
* Comparer la performance des deux programmes
* Choisir le programme le plus performant