

SYSTÈMES EMBARQUÉS I, JOURNAL, TP.01: INTRODUCTION



Haute école d'ingénierie et d'architecture Fribourg
Hochschule für Technik und Architektur Freiburg

Marc Rotten marc.rotten@edu.hefr.ch
Sven Rouvinez sven.rouvinez@edu.hefr.ch
T2-a
Fribourg, September 17, 2018

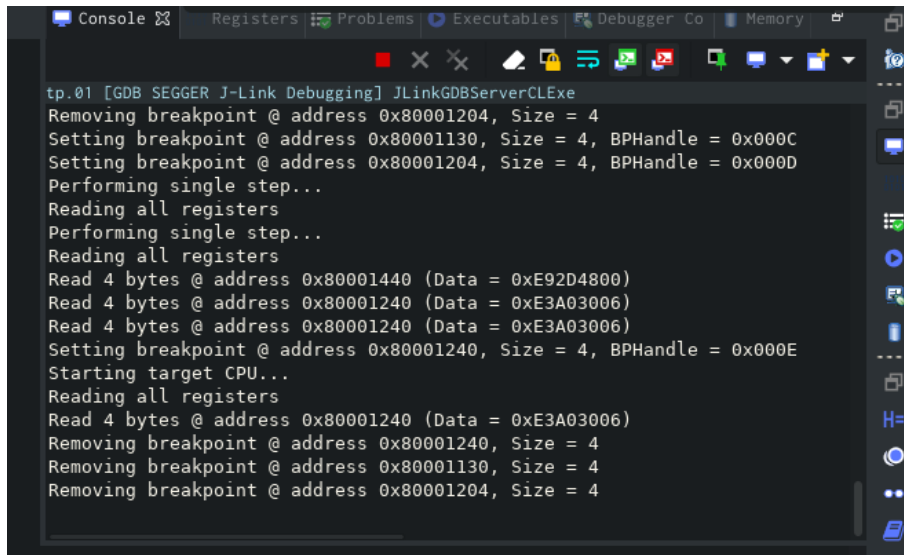


Figure 1 – Taille variables debugger

1 Heures de travail

4 Heures

2 Synthèse

Ce premier TP nous permettra de préparer notre environnement de travail qui comprend une gestionnaire de versions GIT, un IDE eclipse photon et l'utilisation de la programmation croisée nous demande d'utiliser une beaglebone afin de pouvoir interagir avec nous utilisons une liaison J-TAG

3 Question

Quelle est la taille de chacune des variables ?

- leds: 64 bytes
- gpio_init: 28 bytes
- lut: 4 bytes
- i: 64 bytes
- value: 4 bytes
- banner: 4 bytes
- msg: 4 bytes

Quelle est la taille du code ? Environ 164 lignes pour 3.9KB

Comment procéder pour obtenir ces tailles ? Pour la taille des variables, avec le debugger et la taille du code avec `ls -la` sur GNU/Linux

Où se trouve chaque variable en mémoire (adresse absolue) ?

- value: 0x00F020E0
- gpio_init: 0x8000e200
- banner: 0x8000e21c
- i: 0x800010C0

- lut: 0x800012A4
- leds: 0x00F020E0
- msg: 0x8000e2ac

Nous avons trouvé sur le web que l'on pouvait avec l'opérateur & afficher l'adresse d'une variable: `printf("%p", x)`

Où se trouve le code en mémoire? Dans la heap, voir <https://www.geeksforgeeks.org/memory-layout-of-c-program/>

Est-il possible d'améliorer / d'optimiser le code ? Si, oui comment ? Ayant très peu d'expérience avec C, nous avons pas trouvé comment le rendre plus efficace, nous avons émis une liste d'hypothèses:

- Limiter la taille (au niveau du type choisi) des variables pour économiser de la place
- Contrôler qu'il n'y ait pas de memory leak
- Les boucles while infinie sont dangereuse si tout à coup un problème survient sans pouvoir sortir

Comment fonctionne de la macro "ARRAY_SIZE(x)" Il utilise la taille du en bytes du type choisit du tableau passé en paramètre et la divise avec la taille du type de son premier de son premier élément et permet de connaître le nombre d'élément dans un tableau. Dans ce code, le ARRAY_SIZE est utilisé pour boucler à travers les GPIO

4 Feedback

Les questions étaient relativement évasives et étant donnée que nous avions jamais eu de cours sur GIT ou C, il nous a fallu un peu de temps pour les appréhender. Nos réponses peuvent ne pas être correctes car nous sommes encore en train d'apprendre l'environnement dans lequel nous allons évoluer ces semestres