

Donnons du Lustre aux leds

Programmation et validation de systèmes embarqués

Laboratoire Verimag – Faites de la science !

1 Introduction

Au laboratoire Verimag, nous cherchons à aider les ingénieurs informaticiens à prouver automatiquement que leurs programmes sont corrects.

- La correction des programmes est cruciale, en particulier pour les **systèmes embarqués**, car des vies sont souvent en jeu.
- Mais elle est difficile à cause de l'**explosion combinatoire**. Une bonne illustration de ce phénomène est donnée par l'échiquier de Sissa, comportant 1 grain de riz sur case 1, 2 grains sur la case 2, 4 sur la case 3, 8 sur la case 4, etc. Un tel échiquier comporte $2^{64} - 1$ grains de riz, c'est à dire plus de 18 milliards de milliards de grains, soit l'équivalent de 1000 ans de production mondiale de riz actuellement. Or un programme de 64 cases (ou bits) est un petit programme.

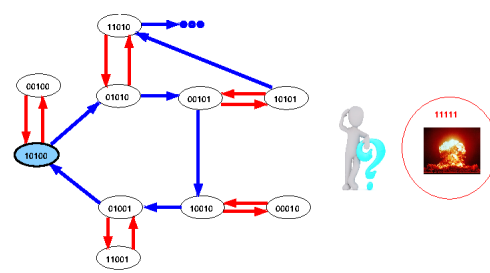
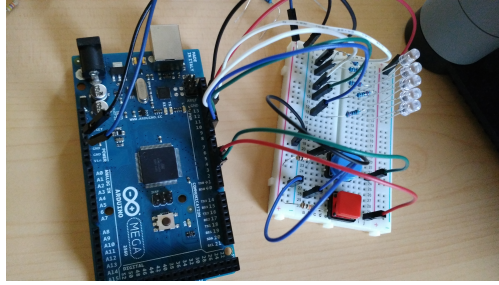
Au laboratoire Verimag, les axes de recherche visant à résoudre ce type de problèmes concernent :

- la conception de **Langages** de programmation empêchant certaines erreurs de programmation ;
- l'**Analyse de programmes** : conception d'algorithmes (des programmes aussi) qui essaient de **prouver** qu'un autre programme est « **correct** ».
- l'automatisation des **tests** : un test qui échoue prouve qu'un programme n'est **pas** correct.

Pour illustrer tous ces axes de recherche, nous allons vous montrer :

1. quelques exemples de programmes écrit en Lustre, un langage inventé au laboratoire et conçu pour mettre en des systèmes embarqués critiques ;
2. comment un tel programme peut-être embarqué sur un système (ici, une carte Arduino) ;
3. comment prouver automatiquement des propriétés sur ces programmes ;
4. comment tester automatiquement ces programmes.

Il s'agit donc de valider un petit système, composé d'un programme Lustre et d'une carte Arduino, comportant 2 entrées, **actionnées** par un bouton rouge et un bouton bleu, et **contrôlant** l'allumage de 5 leds.



Nous proposons plusieurs programmes. Pour chacun d'entre eux, l'effet sur les leds d'une pression sur le bouton bleu ou le bouton rouge est différent. Mais à chaque fois, la propriété de sûreté que l'on veut vérifier est « il est impossible d'allumer les 5 leds ». Le public pourra essayer de résoudre ces petits puzzles, en essayant de déterminer s'il existe une séquence qui amène le système dans l'état dangereux. Nous montrerons ensuite comment des outils peuvent automatiser la réponse à ces petits puzzles.

2 Puzzles

- Puzzle 1
 - Bouton bleu : permute circulairement l'allumage des (jolies) leds
 - Bouton rouge : inverse l'état de la led 5 (éteint → allumé → éteint → etc.)
- Puzzle 2
 - Bouton bleu : idem puzzle 1
 - Bouton rouge : inverse l'état des leds 2 et 4
- Puzzle 3
 - Bouton bleu : idem puzzle 1,2
 - Chaque pression d'un bouton décale la led « courante » : $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 1$.
 - Bouton rouge : inverse l'état de la led « courante/tournante »
- Puzzle 4
 - idem puzzle 3 + le bleu décale la led courante dans le sens inverse : $5 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 5$
- Puzzle 5
 - Chaque pression d'un bouton décale la led « courante » d'un cran ($1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 1$).
 - Bouton bleu : la led courante devient le « xor » des 2 leds autour
 - Bouton red : la led courante devient le « and » des 2 leds autour
- Puzzle 6
 - Bouton rouge : inverse l'état de la led 1
 - Bouton bleu : inverse l'état la led suivant la première led éteinte (en partant de la led1)