

Du Lustre pour des jolies leds

Programmation et validation de systèmes embarqués

Laboratoire Verimag – Faites de la science !

1 Introduction

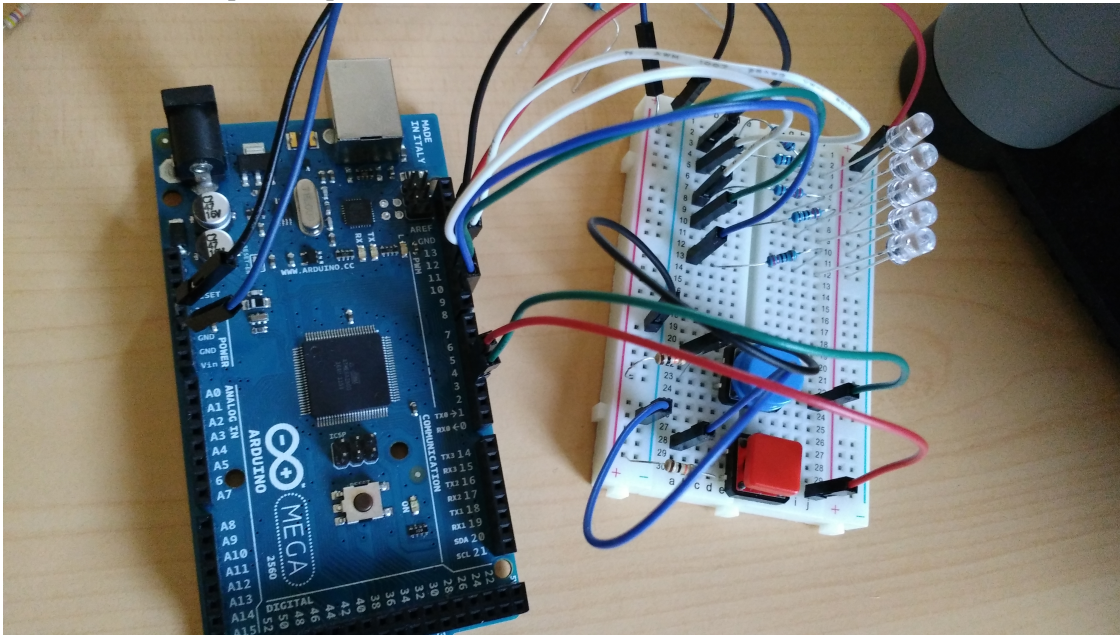
Au laboratoire Verimag, on cherche à aider les ingénieurs informaticiens à prouver automatiquement que leurs programmes sont corrects.

- C'est important, en particulier pour les **systèmes embarqués**.
- Mais c'est difficile (explosion combinatoire ; $2^{64} > 18 * 10^{15}$).

Plusieurs approches :

- **Langages** de programmation (empêcher les programmeurs de faire des erreurs)
- **Analyse de programme**
- Tests automatisés

Nous allons embarquer des puzzles écrits en Lustre sur un Arduino



L'objectif de ces petits puzzles programmés en Lustre est d'allumer toutes les lampes (led) à l'aide de 2 boutons : un rouge, et un bleu.

2 Puzzle 1

2

- Bouton bleu : permute circulairement l'allumage des (jolies) leds
- Bouton rouge : inverse l'état des leds 2 et 4

```
node puzzle1 (red,blue:bool) returns (led1,led2,led3,led4,led5:bool);
let
  led1 = true  -> if blue then pre(led5) else pre(led1) ;
  led2 = false -> if blue then pre(led1) else if red then not(pre(led2)) else pre(led2) ;
  led3 = false -> if blue then pre(led2) else pre(led3) ;
  led4 = false -> if blue then pre(led3) else if red then not(pre(led4)) else pre(led4) ;
  led5 = true  -> if blue then pre(led4) else pre(led5) ;
tel
```

3 Puzzle 2

- Bouton bleu : idem
- Bouton rouge : inverse l'état de la led 5

4 Puzzle 3

- Bouton bleu : idem
- Bouton rouge : inverse l'état d'une led qui circule à chaque pression de boutons

5 Puzzle 4

- Bouton bleu : idem
- Bouton rouge : inverse l'état d'une led qui circule à chaque pression mais dans un sens différent selon la couleur du bouton pressé

6 Puzzle 5

```
true xor false = false xor true = true
false xor false = true xor true = false

true and true = true
true and false = false and true = false and false = false
```

- Bouton bleu => la led courante devient le « xor » des 2 leds autour
- Bouton red => la led courante devient le « and » des 2 leds autour
- la led courante est la première au début, puis la deuxième, etc.

7 Puzzle 6 :

3

- Bouton rouge => inversion de la led 1
- Bouton bleu => allume la led suivant la première led éteinte (en partant de la led1)

```
node puzzle6(red,blue:bool)
returns (led1,led2,led3,led4,led5:bool);
var
  -- Xi = je suis la led qui suit la plus petite led éteinte
  X2, X3, X4, X5 : bool;
let
  X2, X3, X4, X5 = (true, F, F, F) ->
    if not(pre(led1)) then (true, F, F, F) else
    if not(pre(led2)) then (F, true, F, F) else
    if not(pre(led3)) then (F, F, true, F) else
    if not(pre(led4)) then (F, F, F, true) else (F, F, F, F);

  led1 = F -> if red then not(pre(led1)) else pre(led1);
  led2 = F -> if blue and X2 then not(pre(led2)) else pre(led2);
  led3 = F -> if blue and X3 then not(pre(led3)) else pre(led3);
  led4 = F -> if blue and X4 then not(pre(led4)) else pre(led4);
  led5 = F -> if blue and X5 then not(pre(led5)) else pre(led5);
tel
```