

INGÉNIERIE DES SYSTÈMES PROGRAMMÉS INDUSTRIELS

TP1 – Ingénierie des exigences

Pierre-Yves Piriou – EDF

2023-2024

pierre-yves.piriou@edf.fr



TP : Installation de la chaîne d'outils

- Pour l'ingénierie des exigences
 - OpenModelica : <https://openmodelica.org/>
 - CRML-compiler : <https://github.com/p1way/crml-compiler>
- Pour la programmation automate
 - OpenPLC Editor : <https://autonomylogic.com/>
 - (OpenPLC Runtime : https://github.com/thiagoralves/OpenPLC_v3)
 - ScadaBR : <https://github.com/ScadaBR/ScadaBR>
- Pour le test
 - ?
- Pour la programmation HDL
 - FloPoCo : <https://www.flopoco.org/>
 - nvc : <https://github.com/nickg/nvc>
 - GtkWave : <https://gtkwave.sourceforge.net/>
 - (Intel Quartus Lite (+ Max10 device support) : <https://www.intel.fr/content/www/fr/fr/products/details/fpga/development-tools/quartus-prime/resource.html>)

Option : installer le tout dans une machine virtuelle avec VirtualBox (<https://www.virtualbox.org/>)

TP 1.1 : conception d'un contrôleur de feu tricolore



- Etape 1 : spécifier les exigences (comportementales) d'un dispositif de feux tricolores, avec vos mots (30 minutes)
- Etape 2 : échanger vos exigences avec votre binôme et coder le contrôleur conforme aux exigences de votre binôme, dans le langage de votre choix (45 minutes)
- Etape 3 : mise en commun (15 minutes)
- Guide de rédaction du rapport d'expérimentation :
Quelles difficultés avez-vous rencontré dans la spécification des exigences et dans l'implémentation ?
Comment les avez-vous surmontées (ou pas) ?
Avez-vous des pistes d'amélioration concernant votre solution ou son processus de développement ?

N'hésitez pas à inclure dans votre rapport des extraits d'exigences et de code pour illustrer vos propos.

TP 1.2 : conception d'un contrôleur de feu tricolore – la revanche

- Etape 1 : spécifier les exigences (comportementales) d'un dispositif de feux tricolores, avec **CRML** (30 minutes)
 - *Etape 1 bis : traduire en Modelica (graphiquement)*
- Etape 2 : modéliser le contrôleur conforme aux exigences de votre binôme, avec la librairie **StateGraph de Modelica** (45 minutes)
- Etape 3 : mise en commun (15 minutes)



xkcd/2841

« speed limit: 45MPH /
Minimum: 65MPH »

Squelette de la solution :

<https://github.com/p1way/crml-compiler/resources/modelica-libraries/CRML.mo>

↳ CRML/examples/TrafficLight

Fichier CRML : https://github.com/p1way/crml-compiler/resources/crml_tutorial/traffic_light/Spec.crml

Colorateur syntaxique (pour notepad++) :

https://github.com/p1way/crml-compiler/blob/main/resources/notepad_profile/crml.xml

TP 1.3 : conception d'un contrôleur de feu tricolore – la belle

- Complexifier le cas, en considérant maintenant la régulation d'un carrefour.
 - *En CRML textuel, vous pourrez utiliser les « localisateurs spatiaux »*
 - *En CRML graphique (Modelica), vous devrez « mettre à plat » le modèle d'exigence, en instanciant autant d'exigences du même type qu'il y a de feu.*

