

Programmation synchrone

S1 2020–2021, Master 2 Informatique et EIDD 3A

<http://www.irif.fr/guatto/teaching/20-21/progsync>

Adrien Guatto

Bureau 4026, bâtiment Sophie Germain

guatto@irif.fr

Contenu Introduction aux systèmes réactifs. Langages synchrones : programmation (flots-de-données, contrôle, tableaux) et concepts (causalité, compilation en boucle-simple). Notions d'automatique appliquée élémentaire (échantillonnage et contrôle PID). Introduction à l'implémentation des langages synchrones.

Objectifs Ce module offre une introduction à la programmation de systèmes informatiques en interaction continue avec un environnement extérieur, les *systèmes réactifs*, par le biais de langages de programmation spécialisés, les *langages synchrones*.

À la fin du module, vous :

- saurez reconnaître les systèmes réactifs qui vous environnent,
- connaîtrez les principaux problèmes soulevés par la conception et réalisation de systèmes réactifs,
- saurez écrire des programmes synchrones capable de contrôler des dispositifs non-triviaux,
- serez familiarisés avec les mathématiques des systèmes réactifs, y compris des rudiments d'automatique appliquée,
- aurez une idée du fonctionnement général d'un compilateur de langage synchrone.

Prérequis Le module suppose une familiarité avec la programmation en général et le développement sous environnement UNIX en particulier. La connaissance d'un langage fonctionnel comme OCaml ou Haskell est la bienvenue, sans être obligatoire.

Matériel pédagogique Le module ne s'appuie pas sur un manuel particulier. En revanche, *la lecture des notes disponibles sur la page web du module est obligatoire*. Il sera également fait référence à des articles scientifiques dont la lecture est conseillée.

On utilisera le langage universitaire Heptagon [Delaval et al., 2017] aussi bien durant le cours magistral, pour illustrer les principes généraux des langages synchrones, que durant les séances de travaux pratiques. Les autres langages synchrones (notamment d'origine industrielle) seront régulièrement évoqués en cours.

Organisation Chaque semaine comprend 2h de cours magistral et 2h de travaux pratiques.

Séance	Enseignant	Horaire	Lieu
Cours magistral	Adrien Guatto	mardi 10h30–12h30	salle 247E, Halle aux Farines
Travaux pratiques (M2)	Adrien Guatto	mardi 8h30–10h30	salle 2001, Sophie Germain
Travaux pratiques (EIDD)	Alexis Saurin	lundi 10h30–12h30	salle 2003, Sophie Germain

Les séances de travaux pratiques se structurent en deux blocs. Durant la première moitié du semestre, elles vous permettront de prendre en main le langage synchrone Heptagon à travers des exercices de programmation. Ensuite, durant la seconde moitié, elles serviront de soutien à la réalisation du projet.

Planning du cours

29/09 Présentation du module. Introduction aux systèmes réactifs. Présentation d'Heptagon.

- Lecture conseillée : Lee [2008].

06/10 Heptagon : programmation flots de données. Programmes causaux et non-causaux.

15/10 Heptagon : structures de contrôle. Automates hiérarchiques. Transitions faibles et fortes.

- 22/10 Heptagon : structures de données. Tableaux et ordre supérieur.
- 29/10 Présentation du projet.
- 03/11 Synchrone et automatique élémentaire : échantillonnage. Analyse WCET.
- 10/11 Automatique élémentaire : contrôle proportionnel-intégral-dérivé (PID).
- 17/11 Implémentation des langages synchrones : analyse de causalité.
- 24/11 Implémentation des langages synchrones : génération de code en boucle simple.
 - Lecture conseillée : [Biernacki et al. \[2008\]](#).
- 01/12 Introduction aux programmes réactifs hybrides.
 - Lecture conseillée : [Bourke et al. \[2017\]](#), [Benveniste et al. \[2018\]](#).
- 08/12 Révisions et soutien au projet.

Évaluation La note finale est formée pour moitié de la note du projet et pour moitié de la note à l’examen. Les projets seront réalisés en binôme. Chaque binôme est supposé livrer un travail autonome. Vous êtes autorisés et même encouragés à discuter des problèmes que vous rencontrez avec les autres binômes. En revanche, *l’échange de code est interdit* et entraînera l’attribution d’un 0 au projet des binômes concernés.

Informations pratiques

- Rappel : vous avez du 5 au 9 octobre pour finaliser vos inscriptions pédagogiques.
- Si vous rencontrez des difficultés pour installer Heptagon sur votre ordinateur, veuillez vous référer aux instructions sur la page du module.

Références

- Albert Benveniste, Timothy Bourke, Benoît Caillaud, Jean-Louis Colaço, Cédric Pasteur, and Marc Pouzet. Building a hybrid systems modeler on synchronous languages principles. *Proceedings of the IEEE*, 106(9):1568–1592, 2018. doi: 10.1109/JPROC.2018.2858016. URL <https://www.di.ens.fr/pouzet/bib/ieee18.pdf>.
- Dariusz Biernacki, Jean-Louis Colaço, Gregoire Hamon, and Marc Pouzet. Clock-directed modular code generation for synchronous data-flow languages. In *Languages, Compilers and Tools for Embedded Systems (LCTES’08)*, 2008. URL <https://www.di.ens.fr/pouzet/bib/lctes08a.pdf>.
- Timothy Bourke, Francois Carcenac, Jean-Louis Colaço, Bruno Pagano, Cédric Pasteur, and Marc Pouzet. A Synchronous Look at the Simulink Standard Library. In *International Conference on Embedded Software (EMSOFT’17)*. ACM, 2017. URL <https://www.di.ens.fr/pouzet/bib/emsoft17.pdf>.
- Gwenaél Delaval, Léonard Gérard, Adrien Guatto, Hervé Marchand, Cédric Pasteur, Marc Pouzet, and Éric Rutten. The Heptagon synchronous language. <http://heptagon.gforge.inria.fr>, 2017.
- Edward A. Lee. Cyber Physical Systems : Design Challenges. In *International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*. IEEE, 2008. URL <https://www2.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-8.pdf>.