

Programmation synchrone

S1 2024, Master 2 Informatique et EIDD 3A

<https://moule.informatique.univ-paris-diderot.fr/guatto/progsync-m2-eidd-2024>

Adrien Guatto

Bureau 3019, 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 2 heures de cours magistral et 2 heures de travaux pratiques.

Séance	Enseignant	Horaire	Salle
Cours magistral	Adrien Guatto	mardi 11h45–13h45	419C, Halle aux Farines
Travaux pratiques (M2)	Gabriel Scherer	jeudi 08h30–10h30	2003, Sophie Germain
Travaux pratiques (EIDD)	Guillaume Baudart	mardi 16h30–18h30	2027, 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. Vous devez **amener votre ordinateur portable** aux séances de travaux pratiques.

Planning du cours

1. Présentation du module. Introduction aux systèmes réactifs. Présentation d'Heptagon.
 - Lecture conseillée : Lee [2008].
2. Heptagon : programmation flots de données. Programmes causaux et non-causaux.

3. Heptagon : structures de contrôle. Automates hiérarchiques. Transitions faibles et fortes.
4. Heptagon : structures de données. Tableaux et ordre supérieur.
5. Présentation du projet.
6. Applications : génération audio en temps réel.
7. Applications : pendule inversé.
8. Implémentation des langages synchrones : analyse de causalité.
9. Implémentation des langages synchrones : génération de code en boucle simple.
 - Lecture conseillée : Biernacki et al. [2008].
10. Implémentation des langages synchrones : analyse de temps d'exécution pire cas, génération de code temps-réel.

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. L'utilisation d'intelligences artificielles génératives, sous une forme ou une autre (ChatGPT, Copilot, etc.), est également interdite.

Informations pratiques

- Vous devez installer Heptagon sur votre ordinateur portable avant la première séance de travaux pratiques. Si vous rencontrez des difficultés pour installer Heptagon sur votre ordinateur, veuillez vous référer aux instructions sur la page du module.

Références

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>.

Gwenaél Delaval, Léonard Gérard, Adrien Guatto, Hervé Marchand, Cédric Pasteur, Marc Pouzet, and Éric Rutten. The Heptagon synchronous language. <https://gitlab.inria.fr/synchrone/heptagon/>, 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>.