

ALEXIS ECHARD

L'AUTOMATISME INDUSTRIEL EN PRATIQUE

CONCEPTS DE BASE, EXERCICES CORRIGÉS ET MINI-PROJETS

DUNOD

Direction et conception graphiques de la couverture :
Nicolas Wiel et Léa Coma (graphiste)

© Dunod, 2025
11 rue Paul Bert, 92240 Malakoff
www.dunod.com
ISBN 978-2-10-089159-7

TABLE DES MATIÈRES

Préface	XI
---------	----

A

Les bases de l'automatisme industriel

1 ● Introduction aux Automates Programmables Industriels	2
1.1 Qu'est-ce qu'un Automate Programmable Industriel ?	2
1.2 Rôle d'un API	2
1.3 Un peu d'histoire	3
1.4 Les principaux constructeurs	4
1.5 Les domaines d'application	4
1.6 Pourquoi automatiser ?	4
1.7 Comment apprendre	5
1.8 Conseils pratiques pour se lancer	6
1.9 Pourquoi apprendre à programmer un API ?	6
1.10 Questions et exercices	7
2 ● Les types de signaux	8
2.1 Les signaux Tout Ou Rien (TOR)	8
2.2 Les signaux analogiques	9
2.3 Les signaux numériques	9
2.4 Questions et exercices	10

3 ● Les capteurs	12
3.1 Les capteurs TOR	12
3.2 Les capteurs analogiques	14
3.3 Questions et exercices	14
4 ● Les actionneurs	16
4.1 Les actionneurs TOR	16
4.2 Les actionneurs analogiques	18
4.3 Questions et exercices	18
5 ● L'API	20
5.1 Architecture d'un API	20
5.1.1 Les modules d'entrée/sortie (E/S)	20
5.1.2 L'unité de traitement	24
5.1.3 Les mémoires	24
5.1.4 Les modules d'alimentation	24
5.1.5 Les modules de communication	25
5.2 Le fonctionnement d'un API	25
5.2.1 Les modes de fonctionnement	25
5.2.2 Le traitement du programme	25
5.2.3 Les types de démarrage	26
5.3 Choisir un API	27
5.4 Questions et exercices	28
6 ● Les Interfaces Homme Machine (IHM)	30
6.1 Les IHM mécaniques	30
6.2 Les IHM informatisées	30
6.3 Différence entre IHM et SCADA	31
6.4 Questions et exercices	31
7 ● Fondamentaux de la programmation	33
7.1 Qu'est-ce qu'un programme ?	33
7.2 Pourquoi programmer un API ?	34
7.3 Développer un programme	35

7.4	Les logiques combinatoire et séquentielle	36
7.5	Représentation de l'information	36
7.5.1	Organisation en mémoire	36
7.5.2	Numération binaire	36
7.5.3	Conversion au format décimal	37
7.6	Utiliser des variables	38
7.6.1	Qu'est-ce qu'une variable ?	38
7.6.2	Les types les plus courants	39
7.6.3	Représentation des variables en mémoire	39
7.6.4	Nommer ses variables	40
7.7	Questions et exercices	41
8 ●	Tester ses programmes	44
8.1	Les tests en simulation	44
8.2	Les tests sur matériel physique	45
8.3	Questions et exercices	46
9 ●	Les langages de programmation IEC 61131-3	47
9.1	Le langage Ladder	48
9.1.1	Points forts et faiblesses	49
9.1.2	Les réseaux	49
9.1.3	Ordre d'exécution des réseaux	50
9.1.4	Éléments du langage	50
9.1.5	L'affectation	52
9.1.6	Les opérations logiques	53
9.1.7	Les opérations arithmétiques	54
9.1.8	Les comparaisons	55
9.1.9	Autres fonctions	55
9.2	Le langage Schémas à Blocs Fonctionnels	56
9.2.1	Points forts et faiblesses	57
9.2.2	L'affectation	57
9.2.3	Les opérations logiques	57
9.2.4	Les opérations arithmétiques	58
9.2.5	Les comparaisons	59
9.2.6	Autres fonctions	60

9.3	Le langage <i>Sequential Function Chart</i>	60
9.3.1	Points forts et faiblesses	61
9.3.2	Rappel sur les éléments du langage	61
9.3.3	Les règles d'évolution	62
9.3.4	Les divergences	62
9.3.5	Les convergences	63
9.4	Le langage Texte Structuré	64
9.4.1	Points forts et faiblesses	65
9.4.2	L'affectation	65
9.4.3	Les opérateurs	65
9.4.4	Les opérations logiques	66
9.4.5	Les opérations arithmétiques	67
9.4.6	Les opérations de comparaison	68
9.4.7	Les conditions	68
9.4.8	Les boucles	70
9.4.9	Les fonctions	71
9.5	Questions et exercices	72

B

Apprendre à programmer un API

10 ●	Découverte de TIA Portal	76
10.1	Qu'est-ce que TIA Portal ?	76
10.2	Installation	77
10.3	Installation de TIA Portal	77
10.4	Installation de PLCSIM	78
10.5	Travailler avec TIA Portal	79
10.6	Créer un projet	79
10.7	Créer une configuration matérielle	81
10.7.1	Ajouter un API	82
10.7.2	Configurez l'adresse IP	87
10.8	Déclarer ses variables	88
10.8.1	Les zones mémoires	89

10.8.2	La table des variables	89
10.8.3	Lien entre adresses mémoire et E/S physiques	93
10.9	Questions et exercices	94
11 ●	Premiers pas en Ladder	96
11.1	Créer une instruction de base	96
11.2	Compiler vos programmes	101
11.3	Exemple 1 : Pilotage d'un moteur avec un auto-maintien	102
11.4	Exemple 2 : Pilotage d'un moteur avec des bobines SET/RESET	106
11.5	Exemple 3 : Pilotage automatique d'un ventilateur	107
11.6	Exemple 4 : Pilotage automatique d'un ventilateur avec retards	108
11.7	Exemple 5 : Comptage de pièces métalliques sur un convoyeur	111
11.8	Questions et exercices	114
12 ●	Simuler un API avec PLCSIM	117
12.1	Qu'est-ce que PLCSIM ?	117
12.2	Lancement de PLCSIM	117
12.3	Mise en ligne avec l'API	122
12.4	Passage en mode visualisation	122
12.5	Forcer des variables	124
12.6	Questions et exercices	131
13 ●	Développer en Schéma à Blocs Fonctionnels	132
13.1	Exemple 1 : Pilotage d'un moteur avec un auto-maintien	133
13.2	Exemple 2 : Pilotage automatique d'une vanne de régulation	137
13.3	Questions et exercices	140
14 ●	Structurer ses programmes	144
14.1	Les fonctions (<i>Function</i> ou FC)	144
14.2	Les blocs fonctionnels (<i>Function Block</i> ou FB)	148
15 ●	Développer en langage Texte Structuré	154
15.1	Exemple 1 : Pilotage d'un moteur avec un auto-maintien	157
15.2	Exemple 2 : Pilotage automatique d'une vanne de régulation	160

15.3 Exemple 3 : Contrôle d'accès d'un local	162
15.4 Questions et exercices	165
16 ● Introduction au langage SFC	171
16.1 Exemple 1 : Séquence simple	171
16.2 Questions et exercices	178
C Développer, tester et déployer un projet complet – Mini-projet : refroidissement d'une salle serveur	
17 ● Présentation du projet	184
17.1 Avant de commencer	184
17.2 Contexte	184
17.3 Composants du système	185
17.4 Principe de fonctionnement	186
18 ● Créer la configuration matérielle	187
18.1 Ajouter l'API	187
18.2 Définir les E/S	188
19 ● Créer un programme linéaire	190
19.1 Lecture des entrées	190
19.2 Conversion de la température	192
19.3 Gestion du refroidissement	195
19.4 Gestion des défauts	198
19.5 Gestion des alarmes	200
19.6 Écriture des sorties	201
19.7 Félicitations !	202
20 ● Tester en mode simulation	204
20.1 Créer une table de forçage permanent	205
20.2 Scénarios de test	206

21 ● Tester sur API physique	213
21.1 Câbler l'API	213
21.2 Configurer l'adresse IP de l'API	215
21.3 Charger le programme	216
21.4 Quelques tests de mise en service	220
22 ● Créer un programme structuré	224
22.1 Créer le FC_LectureEntrees	225
22.2 Créer le FC_ConversionTemperature	228
22.3 Créer le FC_GestionRefroidissement	231
22.4 Créer le FB_GestionDefaults	233
22.5 Créer le FB_GestionAlarmes	237
22.6 Créer le FC_EcritureSorties	240
22.7 Félicitations !	241
23 ● Ajouter une IHM	243
23.1 Fonctionnalités principales	243
23.2 Créer l'IHM	244
23.3 Connecter l'IHM à l'API	247
23.4 Ajouter des éléments graphiques	248
23.5 Créer les variables IHM	261
23.6 Ajouter les variables IHM au programme API	262
23.7 Lier les variables à la vue IHM	265
23.8 Tester l'IHM	278
Index	283

Pourquoi ce livre ?

J'ai découvert le monde de l'automatisme industriel début 2010 en rejoignant la division nucléaire civile du groupe Rolls-Royce. Fraîchement diplômé, j'ai été recruté en tant qu'ingénieur de mise en service. J'intervenais principalement dans des centrales nucléaires en construction pour installer et démarrer des systèmes automatisés de sécurité. Je remercie d'ailleurs sincèrement les recruteurs de l'époque de m'avoir fait confiance en dépit d'un manque évident de qualification pour le poste. Cette expérience m'a plongé dans un domaine que je n'ai plus quitté par la suite et qui m'intéresse toujours autant aujourd'hui.

Apprendre la programmation en général n'est pas simple. Lorsque j'étais étudiant nous avions peu de ressources d'apprentissage à disposition hormis quelques livres. Il fallait se débrouiller seul ou trouver des personnes pour nous aider. De nos jours, on trouve toutes sortes de supports d'apprentissage gratuits ou non (livres, vidéos, formations...) et la difficulté est maintenant de savoir par où commencer.

En revanche, dans le domaine spécifique de l'automatisme industriel, les ressources sont à mon sens plus rares surtout pour les autodidactes. Hormis les formations de qualité (mais parfois coûteuses) dispensées par les différents constructeurs et les parcours académiques il est difficile de trouver des supports structurés, progressifs et pratiques.

L'automatisme me passionne et j'ai pris énormément de plaisir à apprendre par moi-même durant toutes ces années. J'ai été formé par mes employeurs successifs, appris auprès de personnes plus expérimentées, regardé des tutoriels, acheté des livres et bien sûr beaucoup pratiqué sur le terrain. Je souhaite à travers cet ouvrage retranscrire de la manière la plus simple et structurée possible des connaissances durement acquises en insistant sur les pièges les plus courants.

Programmer un automate est une activité ludique et gratifiante. Quoi de plus satisfaisant que de parvenir (enfin) après plusieurs heures de réflexion à piloter un moteur, un ventilateur ou encore une lampe. Un automate vous permet d'agir concrètement sur votre environnement et c'est ce qui le rend si particulier. C'est la raison pour laquelle j'ai voulu orienter ce livre sur la pratique. Les modes d'apprentissage diffèrent d'une personne à l'autre mais je reste convaincu que, dans les domaines techniques, réaliser par soi-même des exemples, des exercices ou des projets permet d'assimiler durablement un savoir.

J'espère que vous prendrez autant de plaisir que moi à programmer les exemples de ce livre et que son contenu vous sera utile quel que soit votre profil.

Contenu

Ce livre offre un point de vue complémentaire à l'enseignement académique ou aux formations des constructeurs en mettant l'accent sur la compréhension et l'accessibilité. Il a été pensé pour introduire les notions de façon progressives en les illustrant par des exemples concrets. Pour en tirer profit je vous recommande donc d'aborder les différents chapitres dans l'ordre.

Les exercices et le mini-projet permettent de vérifier et de mettre en pratique les connaissances acquises tout au long de l'ouvrage. Certains exercices proviennent de nos tests de recrutement. Prenez le temps de les réaliser et si besoin reportez-vous aux corrections fournies.

L'ouvrage est structuré en 3 parties :

- La première introduit les connaissances de base nécessaires à la mise en pratique.
- La deuxième introduit le logiciel TIA Portal de Siemens pour apprendre concrètement à programmer un automate.
- La troisième est un mini-projet complet pour développer, tester et déployer un programme sur un automate physique.

J'ai fait le choix d'utiliser un automate de la marque Siemens et la suite logicielle associée TIA Portal pour la mise en pratique car ce constructeur reste incontournable de nos jours. La plupart des concepts restent néanmoins transposables à n'importe quel autre fabricant.

À qui s'adresse ce livre ?

J'espère par ce livre faire la promotion de l'automatisme industriel et éveiller la curiosité du plus grand nombre de personnes possibles. Les profils suivants n'en demeurent pas moins les plus susceptibles de tirer profit de son contenu :

- Étudiant en filière automatisme industriel ;
- Jeune ingénieur ou technicien en automatisme ;
- Recruteur ou manager d'équipe technique ;
- Professionnel non-automaticien évoluant dans le domaine (chef de projet, responsable technique...) ;
- Toute personne curieuse intéressée par la programmation d'automate.

A

Les bases de l'automatisme industriel

1 • INTRODUCTION AUX AUTOMATES PROGRAMMABLES INDUSTRIELS

Ce premier chapitre est une introduction générale aux Automates Programmables Industriels (rôles, évolutions, intérêts, constructeurs, domaines d'application...). Il explique également en quoi apprendre à programmer des Automates Programmables Industriels est un bon investissement de nos jours et la manière dont on peut démarrer.

1.1 Qu'est-ce qu'un Automate Programmable Industriel ?

Un Automate Programmable Industriel souvent appelé **API** (ou **PLC** pour *Programmable Logic Controller* en anglais) est un dispositif électronique utilisé pour **automatiser** des processus industriels. Il est le **cœur d'un système automatisé** et permet de contrôler des machines ou des procédés en exécutant des instructions programmées.

Ils sont conçus pour être « relativement » simples à installer et à maintenir. Leur polyvalence permet de les employer pour des applications de tout niveau de complexité (du pilotage d'un portail automatisé à une ligne de production de véhicule).

1.2 Rôle d'un API

Le rôle d'un API est de piloter et de surveiller de manière **autonome** un procédé ou une machine industrielle. Pour remplir sa fonction :

- il reçoit des informations de l'environnement à l'aide de capteurs (entrées) ;
- il traite les informations reçues et élabore des commandes selon la logique d'un programme utilisateur (traitement) ;
- il envoie des commandes à des actionneurs qui agissent sur l'environnement (sorties).