ATELIER DE ROBOTIQUE

Pierre Gaucher
Maître de conférences en informatique
au département informatique de Polytech'Tours

Arnaud Puret
Ingénieur de recherche au laboratoire d'Informatique
de l'Université de Tours

Nicolas Monmarché
Maître de conférences en informatique
au département informatique de Polytech'Tours

Mohamed Slimane
Professeur en informatique
au département informatique de Polytech'Tours

DUNOD

Illustration de couverture :
Computer circuit board © Christian Carnol / istockphoto.com

Le pictogramme qui figure ci-contre mérite une explication. Son objet est d'alerter le lecteur sur la menace que représente pour l'avenir de l'écrit, particulièrement dans le domaine de l'édition technique et universitaire, le développement massif du photocopillage.



Le Code de la propriété intellectuelle du 1er juillet 1992 interdit en effet expressément la photocopie à usage collectif sans autorisation des ayants droit. Or, cette pratique s'est généralisée dans les établissements d'enseignement supérieur, provoquant une baisse brutale des achats de livres et de revues, au point que la possibilité même pour les auteurs de créer des œuvres nouvelles et de les faire éditer correctement est aujourd'hui menacée.

Nous rappelons donc que toute reproduction, partielle ou totale, de la présente publication est interdite sans autorisation de l'auteur, de son éditeur ou du Centre français d'exploitation du droit de copie (CFC, 20, rue des Grands-Augustins, 75006 Paris).

© Dunod, Paris, 2010
ISBN 978-2-10-056132-2

Le Code de la propriété intellectuelle n'autorisant, aux termes de l'article L. 122-5, 2° et 3° a), d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective » et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause est illicite » (art. L. 122-4).

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles L. 335-2 et suivants du Code de la propriété intellectuelle.

Avant-propos

Cet ouvrage s'adresse à des étudiants commençant un cycle scientifique et s'intéressant aux domaines de l'électronique, de l'informatique ou de la robotique. La notion d'autonomie en robotique, d'un point de vue énergétique mais aussi décisionnelle, est particulièrement actuelle. L'ouvrage est conçu pour qu'ils puissent réaliser le robot en totale autonomie. Nous utilisons ce support de travail pour initier les étudiants de 1er cycle universitaire (Parcours des écoles d'ingénieurs Polytech) à la notion plus complexe mais stimulante d'intelligence collective. Il s'agit alors de mettre au point des programmes capables d'engendrer des actions collectives des robots (rangement d'objets, déplacements en formation...).

Le robot présenté ici a été conçu à l'occasion d'une exposition sur les insectes artificiels (issue d'une collaboration entre le département informatique de Polytech'Tours et le muséum d'histoire naturelle de Tours) dans laquelle l'intelligence collective des robots était illustrée par une colonie d'une douzaine de robots. Le montage de cette exposition a impliqué des projets d'étudiants de 1er cycle universitaire (Parcours des écoles d'ingénieurs Polytech).

À travers la construction d'un petit robot autonome et programmable, ce livre vise à initier de jeunes scientifiques à plusieurs domaines techniques :

- L'électronique : les choix des composants et les montages sont expliqués, des conseils de réalisation sont également introduit de façon progressive.
- La mécanique : la réalisation des parties mécaniques du robot est détaillée pas à pas. La mise en œuvre ne nécessite pas d'outillage spécialisé.
- L'informatique : l'utilisation d'un microcontrôleur permet d'aborder la programmation de ce type de calculateur et en particulier de sensibiliser le lecteur aux interactions avec les capteurs/actionneurs du robot. L'aspect programmation est tout d'abord abordé en présentant les principes généraux de fonctionnement des microcontrôleurs. Ensuite, la mise en place des outils de programmation sur le poste de travail sont détaillés. Enfin, les différents aspects de la programmation sont introduit par des exemples de complexité croissante et capitalisant à chaque étape les efforts antérieurs. Chaque étape de programmation sert également à tester les différentes réalisations de la partie électronique. À la fin de l'ouvrage, le lecteur dispose d'une bibliothèque minimale mais fonctionnelle de routines de contrôle en langage C.

© Dunod - La photocopie non autorisée est un délit.

Sommaire

PARTIE 1. RESSOURCES MATERIELLES 3

Chapitre 1. Robotique et microcontrôleur 5
1.1. INTRODUCTION 5
1.2. ARCHITECTURE GÉNÉRALE D'UN VÉHICULE MOBILE 5
1.3. SYSTÈME DE CONTRÔLE 7
Chapitre 2. Le microcontrôleur PIC16F876 11
2.1. CARACTÉRISTIQUES GÉNÉRALES 11
2.2. LA MÉMOIRE 17
2.3. LES TIMERS 28
2.4. LES PORTS D'E/S 31
2.5. REGISTRE DE CONFIGURATION DU MICROCONTRÔLEUR 35
2.6. MISE EN ŒUVRE 37
Chapitre 3. Réalisation du robot : électronique carte PIC 41
3.1. ARCHITECTURE GÉNÉRALE 41
3.2. ACTIONNEURS UTILISÉS 42
3.3. CAPTEURS 43
3.4. PROGRAMMATION ICSP 44
3.5. POSSIBILITÉS D'EXTENSION 44
3.6. AFFECTATION DES RESSOURCES MATÉRIELLES DU ROBOT AU MICROCONTRÔLEUR ... 45
3.7. SCHÉMA DE LA CARTE ÉLECTRONIQUE PIC 47
3.8. RÉALISATION DE LA CARTE ÉLECTRONIQUE PIC 49
3.9. CONNECTIQUE DE LA CARTE ÉLECTRONIQUE PIC 51
Chapitre 4. Réalisation du programmateur 53
4.1. FINALITÉS 53
4.2. SCHÉMA DU PROGRAMMATEUR 54
4.3. RÉALISATION DU PROGRAMMATEUR 55
4.4. CONNECTIQUE DU PROGRAMMATEUR 56
Chapitre 5. Réalisation du robot – mécanique 59
5.1. INTRODUCTION 59
5.2. LE CHÂSSIS 60
5.3. LES CAPTEURS 60
5.4. LES MOTEURS 63
5.5. LES ROUES 67
5.6. ASSEMBLAGE FINAL 68

PARTIE 2. RESSOURCES LOGICIELLES 77

Chapitre 6. Environnement de programmation 79
6.1. GÉNÉRALITÉS 79
6.2. ENVIRONNEMENT DE DÉVELOPPEMENT 81
Chapitre 7. Programmation – premiers pas 99
7.1. INTRODUCTION 99
7.2. FAIRE CLIGNOTER UNE LED 103
7.3. EXÉCUTION DU CODE SUR LE MICROCONTRÔLEUR 111
7.4. TEST DU CAPTEUR ARRIÈRE 114
7.5. TEST DES CAPTEURS AVANT 116

PARTIE 3. GESTION DES RESSOURCES DU MICROCONTRÔLEUR ET DU ROBOT 119

Chapitre 8. Utilisation du Timer0 121
8.1. GÉNÉRATION D'UNE PAUSE 121
8.2. CLIGNOTEMENT D'UNE LED (2) 125
8.3. TIMER0 ET GESTION DES INTERRUPTIONS 126
8.4. CLIGNOTEMENT D'UNE LED (3) 128
Chapitre 9. Gestion des moteurs 133
9.1. PRINCIPE DE COMMANDE D'UN SERVO MOTEUR 133
9.2. TEST DES SERVO MOTEURS 134
9.3. CALIBRAGE DES SERVO MOTEURS 136
9.4. CONTRÔLE DE LA VITESSE ET DU SENS DE ROTATION 141
9.5. CONTRÔLE SIMULTANÉ DES SERVO MOTEURS 146
9.6. CONTRÔLE DE LA DURÉE D'UN MOUVEMENT 152
9.7. ENCHAÎNER PLUSIEURS MOUVEMENTS 157
Chapitre 10. Des programmes mieux structurés 159
10.1. FICHIERS D'EN-TÊTE 159
10.2. MISE EN ŒUVRE 161
10.3. GÉNÉRATION DU CODE EXÉCUTABLE 163
10.4. PRÉDÉFINIR DES MOUVEMENTS 165
Chapitre 11. Interaction avec l'environnement 171
11.1. INTRODUCTION 171
11.2. DÉCLENCHER UN MOUVEMENT 171
11.3. ASSOCIER UN MOUVEMENT SELON LE CAPTEUR ACTIVÉ 173
11.4. UN PREMIER COMPORTEMENT DU ROBOT 174
11.5. SE DÉPLACER SANS LIMITE DE DURÉE 175
11.6. UN AUTRE COMPORTEMENT ET D'AUTRES MOUVEMENTS 177
11.7. SE DÉPLACER EN PRENANT EN COMPTE DES OBSTACLES 179

PARTIE 4. CONCLUSIONS – PERSPECTIVES 183

Chapitre 12. Conclusions - Perspectives 185
12.1. CONCLUSIONS 185
12.2. PERSPECTIVES 186

Annexes 189
Bibliographie 234

Partie 1

Ressources matérielles

Chapitre 1

Robotique et microcontrôleur

1.1. INTRODUCTION

Nous allons aborder la construction d'un véhicule mobile autonome. La forme adoptée pour présenter les différentes étapes de la construction consiste à décrire chaque étape pas à pas, sous la forme d'un guide illustré de montage. Le cahier des charges retenu pour ce véhicule est résumé ci-dessous :

- un véhicule autonome ;
- une mécanique simple et compacte ;
- des capacités de programmation ;
- un environnement logiciel de programmation gratuit ;
- une électronique simple à mettre en œuvre ;
- un coût d'environ 80 €.

À partir de ce cahier des charges fonctionnelles, nous allons maintenant présenter les solutions retenues et leur mise en œuvre.

1.2. ARCHITECTURE GÉNÉRALE D'UN VÉHICULE MOBILE

De façon très générale, un véhicule mobile possède l'architecture fonctionnelle telle qu'elle est décrite par la figure ci-après (cf. Figure **1**).

Le système de contrôle est le « cerveau » de notre véhicule. Il assure la gestion des différents capteurs *extéroceptifs* [1], qui permettent d'obtenir des informations sur l'environnement au sein duquel le robot évolue (présence d'objets, détection

[1] Un capteur extéroceptif permet de doter le robot d'une perception sur son monde extérieur. Cela permet d'adapter le comportement du robot. Les capteurs utilisés sont de nature très variée (capteur TOR – Tout Ou Rien, ultrasonore, infra rouge, caméra couleur ou noir et blanc...).

d'obstacles...). Des capteurs *proprioceptifs* [2] peuvent également être utilisés. Leur fonction permet d'obtenir des informations sur l'état interne du robot (par exemple le nombre de tours de roue lors d'un déplacement). Dans notre cas, ceux-ci ne seront pas mis en œuvre.

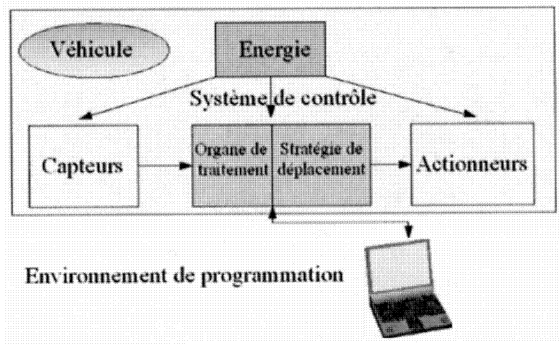


Figure 1 : Architecture générale d'un robot mobile

Le traitement des informations issues des capteurs est réalisé par l'organe de traitement. Les résultats issus de ce traitement permettent d'activer les déplacements du robot conformément à la stratégie de déplacement définie au préalable. Un exemple simple de stratégie de déplacement est le suivant :

- le véhicule avance tout droit tant qu'il ne rencontre pas d'obstacle ;
- si la présence d'un obstacle est détectée, le véhicule effectue une rotation d'un quart de tour vers la droite puis repart tout droit.

Enfin, la mobilité du véhicule est assurée par des actionneurs qui, le plus souvent, sont des moteurs électriques ou des vérins.

Les trois éléments, capteurs, organe de commande et actionneurs, constituent l'ossature du robot lorsqu'il fonctionne en mode autonome.

L'environnement de programmation permet de charger la stratégie de déplacement, et de façon plus générale, le modèle comportemental du robot, au sein du système de contrôle. Cette stratégie de déplacement est codée à l'aide d'un langage de programmation compatible avec le matériel utilisé. Cet aspect de la programmation sera détaillé plus loin.

Nous allons maintenant aborder des aspects plus concrets de la réalisation en commençant par le système de contrôle, suivis des capteurs puis de la motorisation.

---

[2] Un capteur proprioceptif permet de doter le robot d'une perception de son état interne et d'appréhender l'état dans lequel il se trouve. Il peut s'agir par exemple de la mesure de l'autonomie du robot afin de savoir s'il doit trouver une borne de charge pour ses batteries.

1.3. SYSTÈME DE CONTRÔLE

1.3.1. Choix fonctionnel

Le fonctionnement du véhicule nécessite de pouvoir :

- gérer des capteurs en entrée ;
- gérer des actionneurs en sortie ;
- charger un programme qui implémentera la stratégie de déplacement définie au préalable.

Pratiquement, ceci requiert la construction d'une carte électronique capable d'offrir ces fonctionnalités, qui soit simple, compacte, programmable et dont le coût de revient ne sera pas trop élevé. Parmi les solutions disponibles, celles basées sur l'utilisation d'un microcontrôleur sont les plus appropriées.

1.3.2. Spécificité d'un microcontrôleur

Un microcontrôleur contient sous un volume réduit l'ensemble des ressources qui permettent habituellement de contrôler un processus industriel. C'est ce qui le différencie fondamentalement du microprocesseur qui nécessite l'adjonction de composants externes pour pouvoir remplir ces fonctions. Les différences ne s'arrêtent pas à cet aspect des choses. Leurs architectures internes respectives sont organisées selon des principes également bien distincts.

Aussi, les microprocesseurs sont bâtis selon l'architecture définie par *Von Neumann* [3] et décrite ci-dessous (cf. Figure **2**). L'unité de contrôle (UC) ou le processeur assure le traitement des données situées en mémoire. Cette mémoire contient aussi bien des données que les programmes à exécuter. Les échanges de données entre l'UC et la mémoire s'effectuent sur des bus qui permettent d'accéder à une zone mémoire (bus d'adresse) ainsi qu'aux données (bus de données). Les éléments stockés en mémoire représentent aussi bien du code programme que des données.

[3] John Von Neumann : (1903-1957), mathématicien américain d'origine hongroise. Ses principaux travaux concernent la théorie des jeux, des ensembles ainsi que la définition de l'architecture d'un calculateur, dite architecture de Non Neumann. Il a participé à la réalisation du calculateur ENIAC.

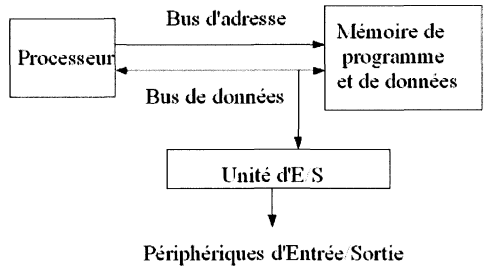


Figure 2 : Architecture de Von Neumann

L'exécution d'une portion de code ou d'une instruction nécessite plusieurs échanges de données : le code de l'instruction ainsi que les données associées à l'instruction devant être exécutés.

Le microcontrôleur peut être conçu selon une architecture interne différente, dite de Harvard (cf. Figure **3**). C'est le cas pour les microcontrôleurs proposés par la société Microchip ou encore certains circuits de chez Atmel.

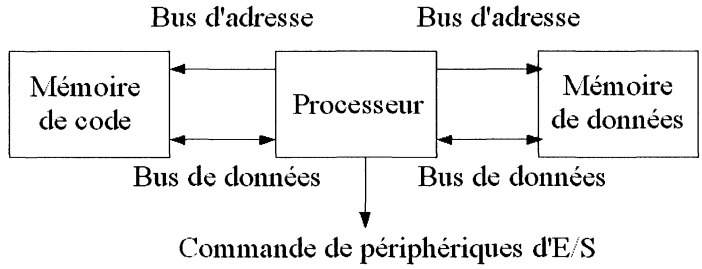


Figure 3 : Architecture de Harvard

Les différences avec l'architecture de Von Neumann résident dans l'organisation de la mémoire. C'est ainsi que les zones mémoire qui permettent de stocker les données et programmes sont maintenant distinctes. Ceci permet alors, simultanément, de pouvoir accéder à la fois aux données et aux instructions. La conséquence directe de cette organisation se traduit par un accroissement de la vitesse de traitement.