

Chapitre 5

Programmation des interruptions et des temporisations

Ce chapitre constitue le cœur de la formation aux systèmes embarqués, car il aborde les périphériques fondamentaux qui permettent au microcontrôleur d'interagir avec son environnement en temps réel. La maîtrise de ces composants est essentielle pour toute application nécessitant une gestion précise du temps, une réactivité aux événements externes ou une acquisition de signaux analogiques. Chaque section suit une approche pédagogique systématique : présentation théorique du périphérique, étude détaillée des registres de configuration, et illustration par des exemples d'applications concrets et progressifs.

5.1. Les entrées-sorties parallèles (GPIO)

Les circuits d'entrées-sorties parallèles, souvent désignés sous l'acronyme GPIO (General Purpose Input/Output), constituent l'interface la plus élémentaire entre le microcontrôleur et le monde extérieur. Ils permettent de connecter des composants simples tels que des LED, des boutons-poussoirs, des afficheurs ou des relais. L'étudiant apprend dans un premier temps à identifier les différents ports disponibles sur le microcontrôleur étudié, ainsi que les broches qui leur sont associées.

La configuration des GPIO repose sur la manipulation de registres spécifiques. Le registre de direction (TRIS pour les microcontrôleurs PIC, ou DDR pour d'autres familles) détermine pour chaque broche si elle fonctionne en entrée ou en sortie. Le registre de données (PORT ou LAT) permet soit de lire l'état logique présent sur une broche configurée en entrée, soit d'imposer un niveau logique (0 ou 1) sur une broche configurée en sortie. Enfin, des registres d'état et de contrôle supplémentaires gèrent des fonctionnalités avancées telles que les résistances de tirage (pull-up) internes, la commutation du niveau logique (toggle) ou la configuration des broches en mode analogique lorsque celles-ci partagent leur fonction avec un convertisseur ADC.

Des exemples d'applications concrètes illustrent cette section : la réalisation d'un chenillard simple avec des LED, la lecture d'un bouton-poussoir avec gestion des rebonds mécaniques (anti-rebond logiciel), ou encore le pilotage d'un afficheur 7 segments. Ces premiers exemples permettent aux étudiants de se familiariser avec la manipulation directe des registres et la lecture des schémas électriques des cartes de développement.

5.2. Les temporisateurs (Timers)

Les temporisateurs, ou Timers, sont des périphériques essentiels pour toute application nécessitant une gestion précise du temps. Ils se présentent généralement sous la forme de compteurs incrémentés par un signal d'horloge, interne ou externe, et déclenchent des événements lorsqu'ils atteignent une valeur prédéfinie. L'étudiant découvre les différentes architectures de Timers disponibles sur le microcontrôleur étudié : Timers 8 bits ou 16 bits, Timers avec ou sans prédiviseur (prescaler), et Timers spécialisés pour la génération de signaux PWM.

La configuration d'un Timer nécessite la maîtrise de plusieurs registres de contrôle. Le registre de sélection de l'horloge détermine la source d'incrémentation (horloge système, oscillateur secondaire, signal externe) et le facteur de division appliqué via le prédiviseur. Les registres de comparaison permettent de définir la valeur seuil à atteindre. Le registre de contrôle du Timer gère l'activation du compteur, le mode de fonctionnement (comptage simple, mode continu, mode capture) et l'effacement automatique du compteur à l'atteinte de la valeur seuil.

Les applications des Timers sont multiples et font l'objet d'exemples progressifs. L'étudiant commence par réaliser une temporisation simple pour faire clignoter une LED à une fréquence déterminée, en utilisant l'attente active (polling) de l'indicateur de débordement. Il aborde ensuite la mesure de durée, par exemple pour mesurer le temps d'appui sur un bouton, puis la génération de signaux PWM (Modulation de Largeur d'Impulsion) pour le contrôle d'un servomoteur ou la variation d'intensité lumineuse d'une LED. Ces exemples illustrent comment les Timers permettent de décharger le processeur des tâches de temporisation rigoureuses.

5.3. Les interruptions

La notion d'interruption est introduite comme un mécanisme fondamental qui permet au microcontrôleur de réagir instantanément à des événements asynchrones, sans avoir à scruter en permanence l'état des périphériques (technique du polling). Contrairement à l'attente active, qui monopolise le processeur, le mécanisme d'interruption permet de suspendre l'exécution du programme principal, d'exécuter une routine spécifique appelée ISR (Interrupt Service Routine), puis de reprendre le cours normal de l'exécution.

L'étudiant apprend à configurer le système d'interruption en manipulant trois types de registres. Les registres de masque (ou d'activation) permettent d'autoriser ou d'inhiber individuellement chaque source d'interruption (débordement de Timer, changement d'état sur une broche, fin de conversion ADC, etc.). Les registres de drapeaux (ou indicateurs) sont positionnés automatiquement par le matériel lorsqu'un événement se produit ; ils doivent être effacés par le programmeur au sein de la routine d'interruption pour éviter un rappel immédiat. Enfin, les registres de priorité, disponibles sur les microcontrôleurs avancés, permettent de hiérarchiser les interruptions lorsqu'elles surviennent simultanément.

La mise en œuvre pratique des interruptions est illustrée par des exemples concrets. L'étudiant programme d'abord un compteur d'événements utilisant une interruption externe sur un bouton, montrant comment l'application principale peut s'exécuter normalement tout en réagissant instantanément à l'appui. Il aborde ensuite l'utilisation conjointe d'un Timer et d'une interruption pour générer des tâches périodiques, fondement de la programmation temps réel. Des précautions essentielles sont également présentées, telles que la gestion des variables partagées entre le programme principal et les routines d'interruption, ainsi que la nécessité de rendre les ISR les plus courtes possibles pour ne pas dégrader la réactivité globale du système.

5.4. Le convertisseur Analogique/Numérique (ADC)

Le convertisseur Analogique/Numérique est un périphérique indispensable pour interfacer le microcontrôleur avec des capteurs analogiques du monde réel, tels que des capteurs de température, des potentiomètres, des capteurs de luminosité ou des capteurs de position. L'étudiant découvre le principe de fonctionnement de l'ADC, basé sur l'échantillonnage et la quantification d'une tension analogique en une valeur numérique binaire.

La configuration de l'ADC implique plusieurs registres de contrôle. Le registre de sélection des canaux détermine quelle broche analogique sera convertie. Le registre de configuration de l'acquisition gère le temps de charge du condensateur d'échantillonnage, paramètre critique pour garantir la précision de la conversion. Le registre de configuration de l'horloge définit la fréquence du convertisseur, qui doit être adaptée aux caractéristiques du microcontrôleur. Enfin, le registre de résultats stocke la valeur numérique (généralement sur 8, 10 ou 12 bits) issue de la conversion.

Les exemples d'applications permettent de mettre en œuvre l'ADC selon différentes stratégies. L'étudiant commence par réaliser une conversion simple en mode polled, attendant activement la fin de la conversion pour lire le résultat, par exemple pour mesurer la position d'un potentiomètre et ajuster la luminosité d'une LED. Il implémente ensuite une version plus avancée utilisant l'interruption de fin de conversion, permettant au programme principal de traiter d'autres tâches pendant que l'acquisition analogique s'effectue en arrière-plan. Des applications plus complexes, telles que l'acquisition d'un signal audio basse fréquence ou la mise en œuvre d'un filtre numérique simple, peuvent être abordées selon le niveau du public et le temps disponible.

5.5. Synthèse et mise en application intégrée

En fin de chapitre, une séance de synthèse permet de consolider l'ensemble des acquis en réalisant une application intégrée combinant plusieurs périphériques. Par exemple, les étudiants peuvent concevoir un système où un capteur analogique (ADC) mesure une grandeur physique, un Timer génère une interruption périodique pour l'échantillonnage, des entrées-sorties parallèles commandent des indicateurs lumineux, et une interruption externe permet de basculer le mode de fonctionnement. Cette approche intégrée prépare les étudiants

Université Djilali BOUNAAMA, Khemis Miliana
مليانة خميس بونعامة جيلالي جامعة
Faculté des Sciences et de la Technologie
والتكنولوجيا العلوم كلية



aux développements plus complexes des chapitres suivants, notamment ceux relatifs aux interfaces de communication.