

Synthèse sur la programmation des ports 2/2**Durée : 1h50**

Objectifs	<ul style="list-style-type: none"> - Mettre en œuvre dans un même projet les principaux périphériques d'un microcontrôleur. - Mettre en œuvre la liaison série sous sa forme la plus simple : protocole de niveau liaison, codage en bande de base, transport avec des niveaux TTL. - Mettre en œuvre le port parallèle. - Mettre en œuvre le Timer.
Prérequis	<ul style="list-style-type: none"> - Le cours ATmega chapitre USART, Ports, Timers - Video d'introduction
Ressources à disposition	<ul style="list-style-type: none"> - PC station de travail sous Windows 10 - Carte Arduino Uno équipée d'un microcontrôleur ATmega328 - Document Le microcontrôleur ATmega328.pptx - Document Memento Représentations.pdf - Document Memento ATmega328.pdf - Barre 6 LEDs
Capacité / compétences visées	<ul style="list-style-type: none"> - S5 Solutions constructives des systèmes d'information - S5.1 Circuits Programmables complexes - S7 Réseaux, télécommunications et modes de transmission - S7.3 Protocoles de bas niveau, Synchrone/asynchrone, half/full duplex, bipoint/multipoints, ...

1 Mise en situation

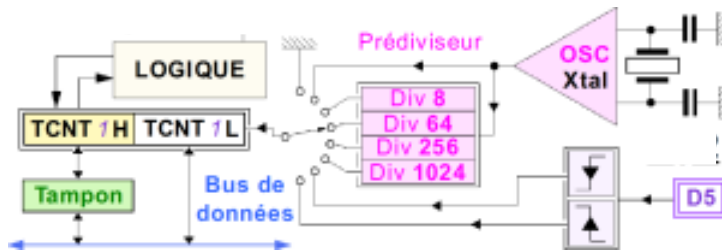
Dans les activités précédentes vous avez mis en œuvre des composants périphériques (Ports, Timers, Liaison série) en manipulant directement leurs registres. Dans cette activité vous allez programmer à la fois l'USART, le Timer, et le port parallèle dans l'objectif de réaliser un jeu simple.

La séance précédente focalisait sur l'USART et le port parallèle. Vous avez réalisé un jeu qui demande à l'utilisateur de convertir en hexadécimal, un nombre affiché binaire par 4 digits.

Dans cette séance introduirez la gestion du timer afin d'estimer le niveau de rapidité du joueur.

2 Éléments de cours

Schéma bloc du timer1 :



L'entrée du timer est soit l'horloge système éventuellement prédivisée, soit une entrée externe. Si l'entrée est externe, ce sera obligatoirement le bit 5 du port D. Le choix interne ou externe se fait par programmation.

Le compteur TCNT est sur 2 octets. Cela signifie qu'il compte sur 2 octets. Il peut donc compter jusqu'à $2^{16}-1$ soit 65535. On dit que le timer1 de l'ATmega328 est un timer 16 bits. Ce registre de comptage doit être lu en deux fois, le poids faible TCNT1L en premier, le poids fort TCNT1H ensuite, car un microcontrôleur 8 bits ne peut pas le lire en une seule fois. Dans notre programme nous le lisons pourtant en une seule fois dans la ligne "tcnt = TCNT1;" mais le compilateur est prévu pour cela.

Le bloc "LOGIQUE" règle les conditions de remise à zéro lorsqu'on atteint un seuil prédéfini (les registres OCR1A et OCR1B pour lesquels on peut calculer un $N=f_e/2f_s$ pour générer une fréquence en sortie).

Les registres de comparaisons OCR1A et OCR1B ne sont pas représentés sur ce schéma.

3 Travail à faire

3.1 Récupération du programme

Clonez ou mettez à jour la librairie <https://github.com/DaKprofSNir/BinMind> en utilisant Github Desktop vers votre espace personnel.

3.2 Test du timer1

Lancez l'environnement Arduino en ouvrant le fichier d'extension '.ino' du programme timer1.

Ce programme montre l'utilisation du timer1 en simple compteur de temps, l'horloge d'entrée étant l'horloge système à 16MHz de l'Arduino UNO.

Téléversez puis ouvrez le moniteur série pour afficher ses résultats.

Le programme commence par afficher les fréquences et périodes comptées par ce timer 16 bits. Décochez "Défilement automatique" pour avoir le temps de lire les résultats de ces calculs.

3.3 Mise en œuvre du timer1 en simple compteur de temps

Repérez les 4 lignes d'initialisation du timer1.

On bloque le timer, on le paramètre, puis on le débloquent.

Déterminez la fréquence qui sera appliquée à l'entrée du compteur (après le prédiviseur) :

.....

Déterminez le quantum de temps que représente une impulsion d'entrée ($T=1/F$) :

.....

Calculez combien de quantums un compteur 16 bits peut compter avant de déborder :

.....

Calculez combien de secondes ce compteur 16 bits peut compter avant de déborder :

.....

Observez dans loop(), les trois lignes qui récupèrent le temps écoulé pour en faire le total. Expliquez pourquoi il faut faire cette récupération avant 4 secondes.

.....

3.4 Amélioration du programme BinMind

Lancez l'environnement Arduino en ouvrant le fichier d'extension '.ino' du programme **BinMind2**.

L'onglet 'montage' explique comment connecter la barre LEDs, connectez-là.

Ce programme est encore incomplet, vous allez calculer les valeurs à placer dans les différents registres.

| //----- Initialisation du timer1 mode normal -----

Terminez l'initialisation du timer en vous basant sur le programme d'exemple timer1.

| //---- lire le compteur avant qu'il ne déborde (son max est 65535)

Terminez la totalisation du temps écoulé en vous basant sur le programme d'exemple timer1.

3.5 Test du programme

Téléversez, lancez, et testez votre programme BinMind2.

4 Conclusion et bilan

Que pensez-vous avoir appris ou approfondi plus particulièrement dans cette activité ?

.....

.....

.....

.....

.....

5 Questions facultatives

À la fin de chaque jeu, la barre diminue jusqu'au niveau calculé du joueur.

Vous constatez qu'il est très difficile d'atteindre le mode expert.

Vous pouvez modifier les valeurs seuils pour rendre la progression plus réaliste.

.....

.....

.....

.....

.....

.....