

## Les Bus : Le Bus PC 104

### Conception, Réalisation et Pilotage d'une carte PC104 multifonctions.

**But du TP : Interfaçage au bus PC104 et développement logiciel sur base PC104.**

Au cours de ce TP, nous allons concevoir et réaliser une carte PC104 multifonctions. Celle-ci mettra en oeuvre un circuit PIO Intel 8255 et un triple Timer programmable Intel 8253/8254.

L'élément de base sera une carte de développement P900 de **Microcomputer Systems, Inc®**, qui regroupe sous une forme pratique les circuits de base de l'interface au bus PC 104 : tampon de données, logique de décodage ainsi qu'une zone de "wrapping" disponible pour placer des circuits spécifiques.

Après réalisation et mise au point du schéma, il faudra mettre en place les circuits sur la zone de wrapping et réaliser le câblage, il s'agira enfin de réaliser des fonctions d'entrées/sorties simples à partir des circuits 8255 et 8253/8254 de cette carte en utilisant l'outil de développement OpenWatcom

#### Matériel :

Composants : supports à wrapper, 8255, 8253/8254, 7406, leds, interrupteurs DIL, résistances et réseaux de résistances.

Multimètre, Oscilloscope

Unité centrale PC104 **ARCOM®** 386EX, Carte support pour UC 386EX , Carte prototypage wrapping PC104 P900, carte intermédiaire à insérer entre l'UC et la carte prototype, pour éviter les court-circuit.

#### Documentation :

Polycopié de cours ISA/PC104

Spécification du 8255 et 8253/8254, et des circuits d'interface TTL-LS.

Manuels des cartes P900, 386EX.

#### 1a) Partie Matérielle - Conception:

La carte multifonction devra comprendre :

- **Un PIO type 8255** dont les broches PA0 et PA1 piloteront respectivement une LED rouge LD0 et une LED verte LD1 en fournissant un courant d'environ 10mA. Les broches PC0 et PC1 serviront à lire l'état ouvert ou fermé de deux interrupteurs DIL SW0 et SW1.
- **Un TIMER type 8253/8254** dont un des étages servira à générer un signal rectangulaire de fréquence variable, par division d'une horloge externe ou interne à la carte. La validation de cette fonction sera faite par l'intermédiaire de la broche PA7 du PIO.

Etablir un schéma de principe **complet** destiné au câblage de la carte. Ce schéma devra pouvoir impérativement être utilisé par une personne extérieure à votre binôme et toutes les informations doivent y figurer. Les noms de signaux doivent correspondre à ceux utilisés sur la carte P900. Etablissez également une liste de pièces du matériel nécessaire.

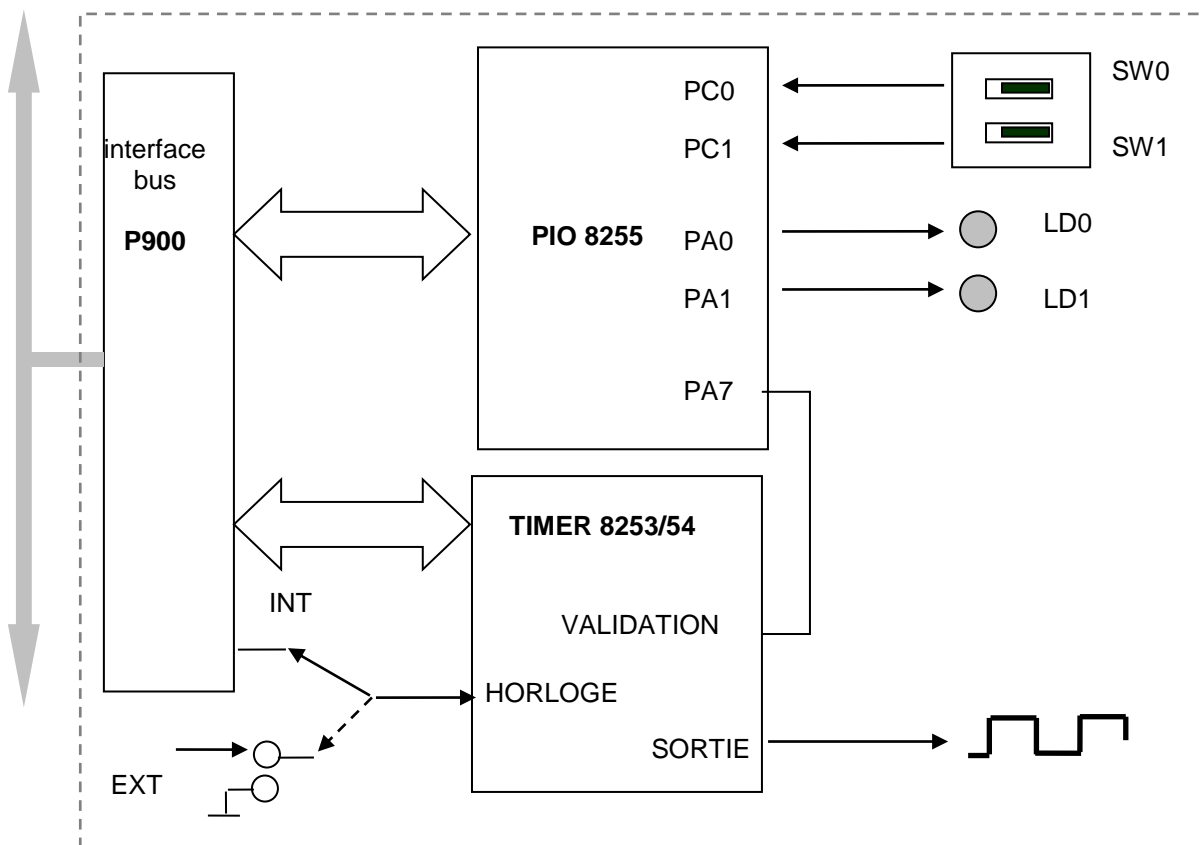


*La qualité de cette partie du travail sera prise en compte séparément dans l'évaluation de votre travail de TP.*

Les adresses des circuits seront fixées comme suit : 8255 : **0300h**      8253/8254 : **0310h**.



*Attention, contrairement aux exemples étudiés en cours, l'espace IO pris en compte est basé ici sur un schéma adressage sur 16 bits*




### 1b) Partie Matérielle - Réalisation:

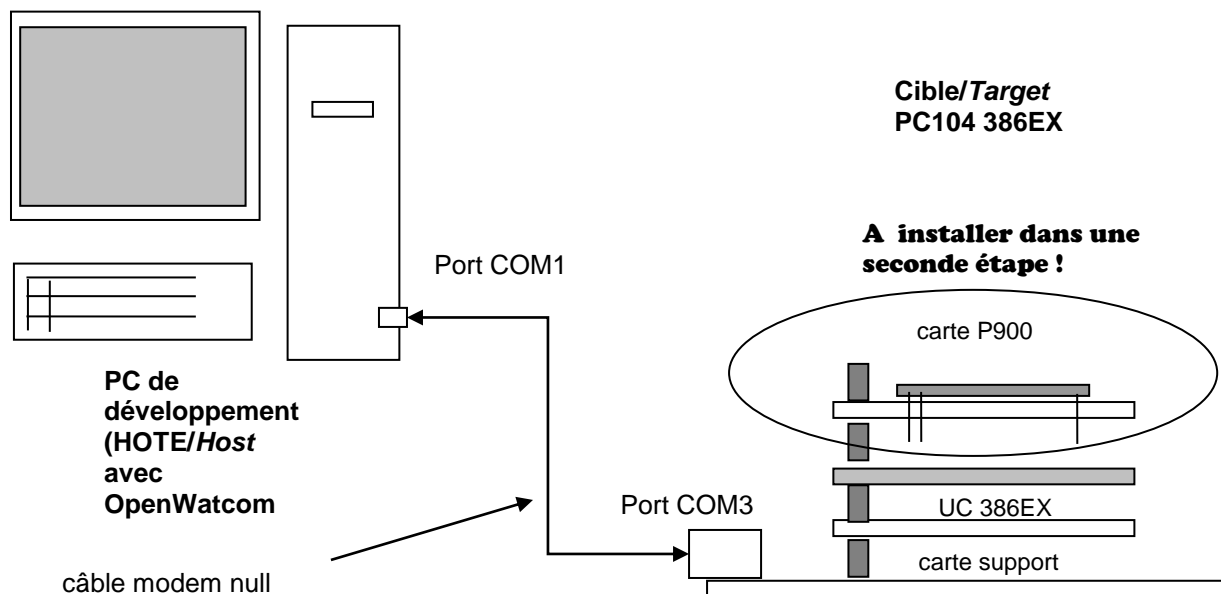
Réaliser le câblage par la technique du wrapping (connexions enroulées). Adopter un code de couleur pour faciliter la vérification, par exemple :

bleu	bus de données
rouge	+5V
noir	GND
vert	signaux de contrôle

Vérifier la conformité de votre réalisation par rapport à votre schéma par "sonnage" de chaque connexion au multimètre.

 Cette étape est indispensable lors du test d'un prototype. Une erreur de câblage peut non seulement entraîner le non fonctionnement de votre montage, mais également aboutir à la destruction du matériel de développement (386EX).

## 2a) Partie logicielle – Prise en main de l'Outil de développement




Le schéma ci-dessus représente l'ensemble de la station de développement que vous allez utiliser.

 *Dans un premier temps, n'installez pas encore votre carte prototype!*

Alimentez la carte 386 EX sur **5V CC** et connectez vous via une session Hyperterminal. Le système d'exploitation résident ROM-DOS, une version embarquée de DOS, s'annonce et vous permet de taper quelques commandes (**DIR** liste les fichiers résidents dans la mémoire Flash de la carte, vue comme le disque C: .

Utilisez le tutorial fourni "**premier projet DOS.pdf**" (cible 386EX), pour éditer, compiler et télécharger un premier programme permettant à votre cible d'afficher le sempiternel "*Hello, world!!*" sur l'écran de l'émulateur de terminal.

 *A ce stade, vous êtes en mesure de réaliser le programme de pilotage de votre carte prototype.*

*Installez-la et vérifiez qu'elle ne perturbe pas le fonctionnement du programme précédent. Si c'est le cas, il y a une erreur de conception ou de câblage.*

## 2b) Partie logicielle – Réalisation du programme de pilotage

Pour piloter les circuits 8255 et 8253/54, il faut tout d'abord accéder à leurs registres. La première étape consiste donc à réaliser la carte mémoire (*memory map*) de ces registres dans la zone IO. Aidez-vous pour cela de la documentation des circuits et tenez compte de l'adresse de base définie plus haut.

L'accès s'effectue ensuite très simplement grâce aux fonctions **inp()** et **outp()** documentées dans le tutoriel et l'aide en ligne de Borland C++.

Réalisez un sous-programme d'initialisation configurant les circuits pour les fonctions demandées, puis écrivez une fonction principale réalisant une des opérations suivantes en fonction d'un choix entré par l'opérateur (un chiffre tapé au clavier) la sortie d'un mode choisi se fera par l'appui sur **ESC**: Travaillez de façon modulaire et mettez au point progressivement vos fonctions. Commentez!

**LEDS** : afficher une séquence binaire sur LED0 - LED1, incrémentation à chaque appui sur une touche.

**INTERS** : à chaque appui sur ENTER, le contenu des interrupteurs est recopié sur les leds.

**TIMER** : générer un signal dont la fréquence sera un sous multiple (facteur de division entré par l'opérateur) de la fréquence du signal d'entrée.

**SCOPE** : Réaliser un programme adressant la carte, en écriture et en lecture, pour pouvoir visualiser à l'oscilloscope les signaux du bus ISA (CS, IOR, IOW, BCLK .... Capturez les chronogrammes obtenus.