

Le clavier PC

et son interface



La connaissance des caractéristiques du clavier PC et de son interface permet d'envisager deux types d'applications : remplacer un clavier matriciel par un clavier PC dans un montage microcontrôlé ou bien encore utiliser l'interface clavier du PC pour entrer des données dans le PC. Cet article se propose de faire découvrir de façon expérimentale le fonctionnement du clavier.

Choix d'un clavier

Lorsqu'un montage microcontrôlé nécessite l'utilisation d'un clavier, on se tourne généralement soit vers un clavier à point commun, soit vers un clavier matriciel.

Les claviers à point commun sont utilisés dans le cas où le nombre de touches nécessaires est faible. En effet, ce type de clavier nécessite une ligne d'entrée du microcontrôleur par touche de clavier. Il est souvent réalisé avec des boutons poussoirs indépendants (avec des touches de type D6 ou autre). Le programme à écrire pour gérer un tel clavier est des plus simples, puisqu'il suffit de lire le niveau logique présent sur chacune des entrées du microcontrôleur. Signalons qu'un tel programme peut fort bien gérer l'appui simultané sur plusieurs touches pour augmenter les possibilités du clavier.

Les claviers matriciels les plus fréquemment utilisés dans les montages publiés sont ceux de type SECME à 12 ou 16 touches. L'avant-

tage du clavier matriciel est le nombre réduit de lignes d'entrée/sortie utilisées sur le microcontrôleur : 7 pour un clavier 12 touches et 8 pour un clavier 16 touches. La diminution du nombre de lignes nécessaires se paie par une gestion logicielle un peu plus complexe que dans le cas du clavier à point commun.

Lorsqu'un nombre de touches plus important s'avère indispensable dans l'application envisagée, on peut alors raisonnablement s'orienter (vu son prix) vers le choix d'un clavier PC quitte à ne pas utiliser toutes les touches disponibles.

Tout ce qui est écrit dans cet article ne concerne que les claviers de type AT/PS2 et n'est pas valable pour les claviers des premiers PC (type XT).

Connectique du clavier

Les premiers claviers type AT étaient livrés avec une fiche DIN 5 broches alors que ceux de type PS2 sont livrés avec une fiche mini DIN 6 broches. Le protocole de communi-

cation reste le même dans les deux cas et l'on pourra utiliser indifféremment l'un ou l'autre. D'un point de vue pratique, il sera peut-être plus facile de trouver, pour réaliser le montage, une embase femelle DIN 5 broches pour circuit imprimé, ce qui permettra d'utiliser directement un ancien clavier ou encore un clavier PS2 avec un adaptateur DIN5/PS2 que l'on peut trouver partout.

La **figure 1** indique le branchement du clavier suivant le type de connecteurs (chaque connecteur est vu de face).

Le **tableau** ci-après donne la correspondance entre les broches des connecteurs et les noms des signaux :

N°	DIN 5 broches	Mini Din 6 Broches
1	Horloge	Données
2	Données	Non connecté
3	Non connecté	Masse
4	Masse	+5V
5	+5V	Horloge
6		Non connecté

Comme on peut le voir dans ce tableau, le clavier doit être alimenté : contrairement aux petits claviers matriciels cités ci-avant et qui sont de type passif, le clavier PC contient une circuiterie électronique de type numérique, d'où l'alimentation en 5V. Elle est fournie par le système hôte (en principe le PC... ou la carte à microcontrôleur dans notre cas). On appellera "micro" le système hôte. Compter quelques dizaines de mA nécessaires au fonctionnement du clavier (il contient lui aussi un microcontrôleur, quelques LED, etc.). Avec les claviers testés, on a relevé des intensités de l'ordre de 40 mA, les LED étant éteintes. Les deux autres lignes (Horloge et Données) seront les seules à être connectées au microcontrôleur, ce qui épargne bien des ressources matérielles sur celui-ci.

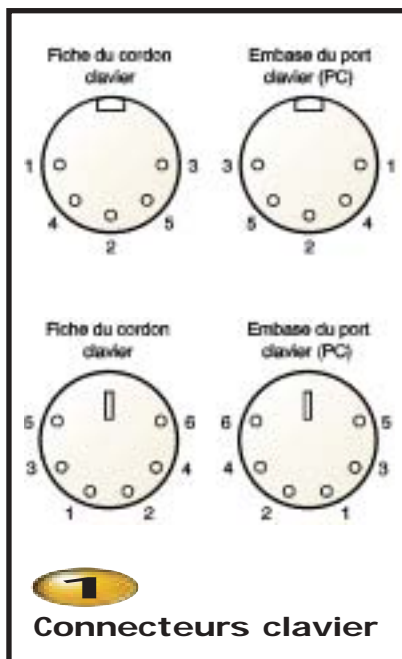
La transmission d'informations entre le clavier et le micro est donc de type sériel mais se fait de façon synchrone.

La communication entre le clavier et le micro est bidirectionnelle mais pas simultanée bien sûr, puisque les deux seules lignes de communication sont communes aux deux systèmes. Les lignes Données et Horloge peuvent être mises au niveau haut ou bas par le clavier ou par le micro. Pour que ceci puisse être réalisé sans conflit, il faut, qu'à chaque extrémité, les lignes soient à collecteur ou drain ouvert avec résistances de rappel au +5V. La patte d'entrée/sortie choisie sur le microcontrôleur doit pouvoir être configurée ainsi. Avec un 68HC11, on pourra utiliser le Port C (en mettant le bit 5, CWOM du registre PIOC à 1).

On pourrait aussi utiliser le Port D (en mettant le bit 5 DWOM du registre SPCR à 1). Nous allons commencer par étudier le protocole de communication dans le sens clavier vers micro.

Communication clavier → micro

Le clavier peut envoyer des données au micro lorsque les lignes Données et Horloge sont au niveau haut. Si le micro met la ligne Horloge à 0, alors le clavier stocke dans sa mémoire tampon les données à envoyer jusqu'à ce qu'elle soit remise à 1. L'horloge de synchronisation est générée par le clavier. La transmission de données du clavier vers le micro se fait sur 11 bits :

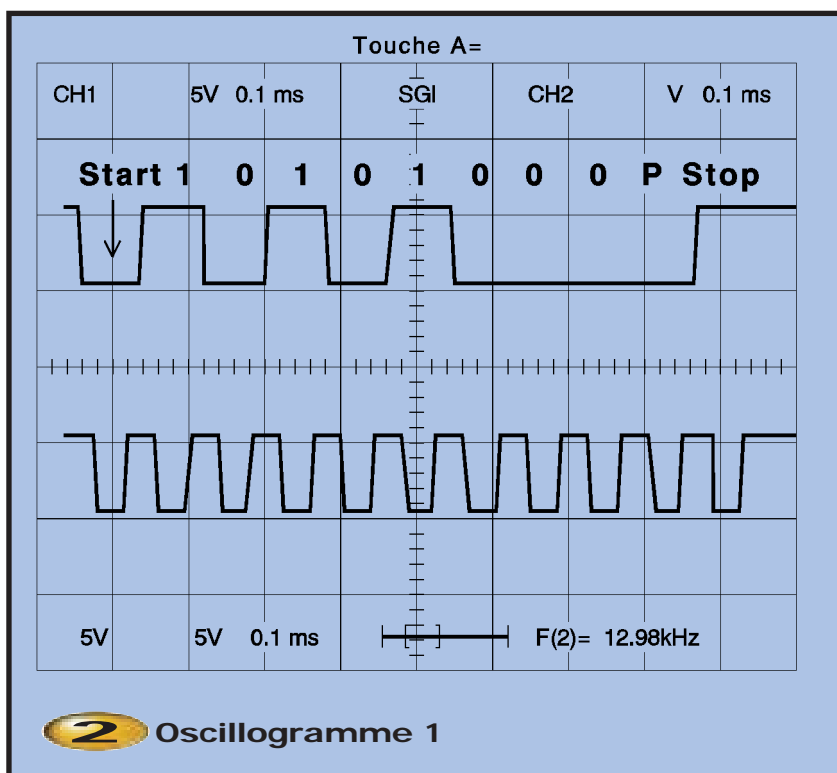


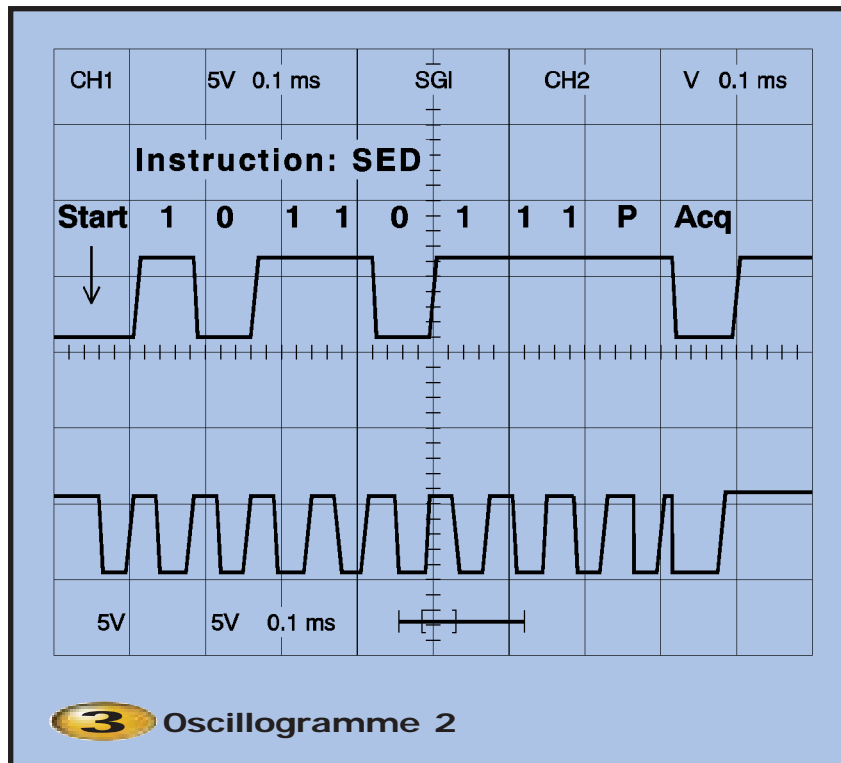
- **1^{er} bit** : bit de start (niveau 0),
- **bits 2 à 9** : octet de donnée (le bit de poids faible en premier),
- **bit 10** : bit de parité (parité impaire),
- **bit 11** : bit de stop (niveau 1).

L'oscillogramme 1 ci-joint montre une trame correspondant à l'appui sur la touche "A" d'un clavier AZERTY. Attention, cet oscillogramme correspondrait à l'appui sur la touche "Q" d'un clavier QWERTY. Cet oscillogramme réalisé sur oscilloscope à mémoire vous permet de voir l'allure des

signaux en situation réelle. En voie 1, affichée dans la partie supérieure, c'est la ligne de Données qui a été mesurée, la ligne Horloge se trouvant dans la partie inférieure sur la voie 2. Les réglages apparaissent dans le haut de l'écran : 5 V/div sur chacune des voies et la base de temps réglée sur 0,1 ms/div. La fréquence du signal d'horloge a été mesurée à 13 kHz environ. Cette fréquence n'est pas standard et peut varier de quelques kHz d'un clavier à un autre. On obtient ici comme octet de donnée 00010101 soit \$15 (hexa). Chaque bit est considéré comme valide après la retombée à zéro du signal d'horloge. Le protocole de transmission se faisant avec parité impaire, le bit de parité est donc à 0 sur cet exemple puisque le nombre de bits égaux à un dans l'octet \$15 était impair.

En fait, la transmission est plus complexe que ce qui vient d'être écrit : le clavier PC peut émettre trois jeux de codes différents (on oubliera ici le jeu n°1 propre aux claviers de type XT qui doivent être en voie d'extinction définitive). Après la mise sous tension ou après un reset, le clavier émettra des codes du jeu n°2 : lorsque l'on appuie sur une touche, le clavier émet alors un code (\$15 pour la touche A). Si on laisse la touche enfoncée, le clavier va continuer d'émettre ce code (\$15) avec un facteur de répétition (que l'on peut programmer). Lorsque l'on relâche la touche, le clavier





émet le code de relâchement \$F0 puis le code de la touche relâchée (\$15 dans cet exemple).

L'appui sur certaines touches du clavier donne lieu à un envoi de codes plus complexes : avec la touche Fin on obtient E0 69 puis E0 F0 69 au relâchement. Voici en prime la suite d'octets envoyés suite à l'appui et au relâchement de la touche Impression Ecran : E0 12 E0 7C E0 F0 7C E0 F0

12 ... un vrai code bancaire !!! Seule la touche pause n'émet pas de code de relâchement.

Lorsque l'on place le clavier dans le jeu de codes n°3, il n'émet pas de code de relâchement (sauf pour les touches CapsLock, Shift, Ctrl gauche et Alt) ce qui diminue le nombre de codes émis et simplifie la gestion des codes en question. On aura certainement intérêt à se placer dans ce jeu

n°3 lorsque l'on voudra utiliser un clavier PC en remplacement d'un clavier matriciel dans un montage microcontrôlé.

Par contre, si on veut utiliser l'interface clavier comme interface de saisie de données, il faudra penser à émettre des codes compatibles avec le jeu n°2.

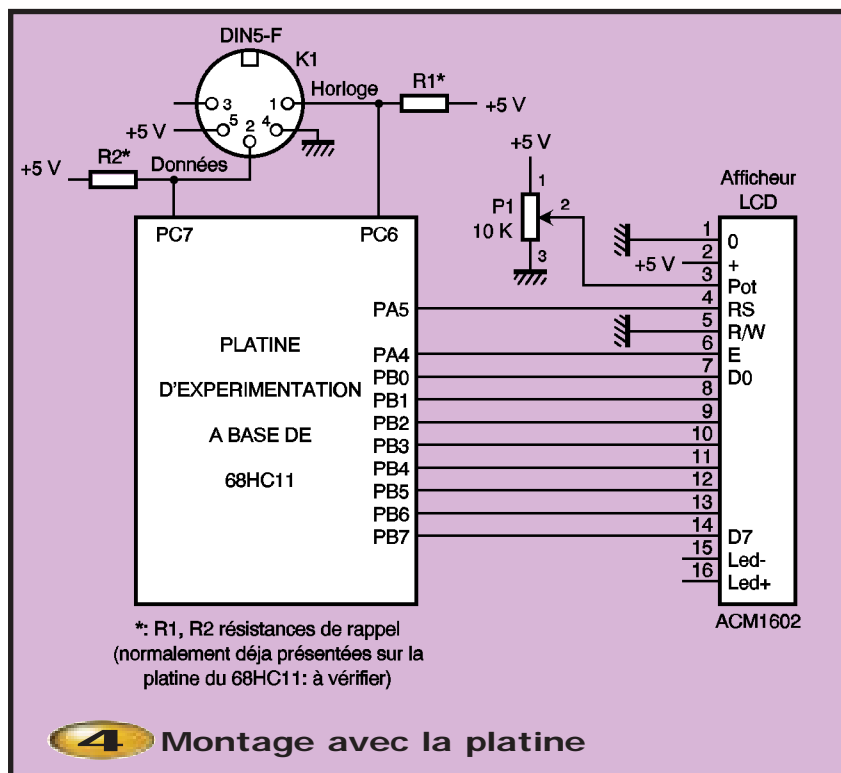
Communication micro → clavier

Il peut être parfois nécessaire d'envoyer des instructions au clavier. Par exemple, lui faire un Reset en début de programme, changer le jeu de codes émis, ou bien encore allumer la LED correspondant à la touche de Verrouillage des majuscules (Caps-Lock). En effet, les LED de contrôle du clavier que l'on voit s'allumer lorsque l'on presse telle ou telle touche sont commandées par instruction provenant du PC.

La procédure est la suivante : les lignes Horloge et Donnée étant initialement au niveau 1, le micro fait tomber la ligne Horloge à 0. Environ 35 µs plus tard, il fait tomber la ligne Donnée à 0. Environ 125 µs après, le micro relâche la ligne d'Horloge (ce qui la ramène à 1 avec la résistance de rappel). Le clavier scrutant de son côté l'état des lignes, la séquence que l'on vient de décomposer est interprétée comme étant une demande du micro pour émettre une instruction. Le clavier commence à générer un signal de synchronisation sur la ligne d'Horloge. On remarque que c'est de nouveau le clavier qui synchronise les communications.

Le micro attend la retombée d'Horloge au niveau 0, ce qui correspond au bit de Start. Il peut alors préparer le bit le moins significatif de l'instruction voulue lorsque la ligne d'Horloge est au niveau 1 et le placer sur la ligne de Donnée. Il est validé lorsque la ligne d'Horloge est revenue au niveau 0. On répète cette procédure pour les 7 autres bits de l'instruction et, une fois de plus, pour le bit de parité (parité impaire). La ligne de Donnée est alors relâchée, ce qui la ramène au niveau 1. Pour signaler la réception de l'octet et du bit de parité (Acquittement), le clavier fait tomber la ligne Donnée à 0 pendant un cycle d'horloge.

L'oscillogramme 2 montre l'émission du code \$ED par le microcontrôleur. Le bit de parité se trouve ici à 1 puisque l'octet \$ED



contenait un nombre pair de bits au niveau 1. Ce code permet de contrôler l'allumage des LED Num Lock, Caps Lock et Scroll Lock. Après émission de ce code, on doit émettre un octet dont les 3 premiers bits de poids faible contrôlent chacune des LED précitées. Le listing source joint (Dec_Cla2.a11) montre une séquence d'allumage et d'extinction de ces LED.

Le tableau ci-après livre quelques-unes des principales instructions que l'on peut envoyer au clavier :

Pour	Instruction (en hexa)	Commentaire
Faire un Reset clavier	\$FF	Le clavier retourne \$FA \$AA et se retrouve dans le jeu de codes N°2
Choisir le jeu de codes	\$F0	Faire suivre d'un octet : \$01, \$02 ou \$03
Contrôler les LED	\$ED	Faire suivre d'un octet contrôlant la ou les LED voulues : \$00 à \$07

Application

Pour illustrer ce qui vient d'être dit et pour aller plus loin dans l'exploration du clavier, nous vous proposons un montage à implanter sur carte d'expérimentation pour 68HC11 (Cf. Electronique Pratique de Juin 97, repris dans le CDRROM 97). Hormis la carte en question, il faudra une embase femelle DIN 5 broches et un afficheur LCD deux lignes de 16 caractères et un ajustable de 10 k Ω pour le réglage du contraste. Ces différents éléments seront montés sur la plaquette d'expérimentation jointe à la carte décrite. La carte pour 68HC11 dans le n°243 (Janvier 2000) pourrait convenir en lui adjoignant une plaquette d'expérimentation pour faciliter l'implantation des quelques composants.

Ce montage permet de décoder, en temps réel, les codes émis par le clavier. Il les affiche sur l'écran LCD au fur et à mesure de leur réception. Les programmes (Dec_Cla2.a11 et Dec_Cla3.a11) sont écrits (en assembleur avec Basic11 de chez Controlord) pour le microcontrôleur 68HC811E2 avec un quartz de 8 MHz mais peuvent être adaptés à d'autres microcontrôleurs car les listings source sont suffisamment commentés. En cas d'adaptation, il faudra veiller aux temps d'exécution car le décodage et l'affichage se font en temps réel.

Dec_Cla2 commence par faire un Reset du clavier en lui envoyant l'instruction \$FF puis attend les données en provenance du clavier. On verra dans cette phase de Reset

les 3 LED de contrôle du clavier s'allumer un court instant, puis le clavier envoie les codes \$FA et \$AA pour indiquer que tout est OK. Ensuite le programme envoie des instructions pour allumer les LED du clavier. Le clavier retourne pour chaque instruction réalisée un octet d'acquiescement \$FA. Puis il reste dans une boucle d'attente de données en provenance du clavier, ce qui permettra de décoder toutes les touches du clavier. Le clavier se trouve ici dans le jeu de codes n°2.



Munis de tous ces renseignements, vous voilà prêts à envisager toutes les applications possibles faisant intervenir le clavier PC (ou son interface) et un microcontrôleur. Bonnes expérimentations !

V. LE-MIEUX

Nomenclature

- carte d'expérimentation pour 68HC11 (voir texte)
- afficheur LCD 2 lignes de 16 caractères (ACM 1602A)
- ajustable de 10 k Ω
- embase femelle DIN 5 broches pour CI
- les résistances de rappel R₁ et R₂ seront égales à 4,7 ou 10 k Ω

Documentation

- "Interface clavier PC pour bus I2C" P.MORIN - Electronique Radio Plans N°575 (Oct 95)
 - Note d'application Motorola AN1723 "Interfacing MC68HC05 Microcontrollers to the IBM AT Keyboard Interface" (cette note présente un montage thermostatique digital alimenté et contrôlé par l'interface clavier d'un PC-AT : schéma + listing).
 - "Interfacing the PC" de C.Peacock à l'adresse : <http://www.geocities.com/SiliconValley/Bay/8302/keybrd.htm>
- Attention** aux sources d'origines étrangères qui décrivent le fonctionnement du clavier QWERTY alors que la grande majorité des claviers "francophones" sont AZERTY !

Logiciels et fichiers

Dec_Cla2.A11 , Dec_Cla3.a11 et Mes_Def.A11 plus Dec_Cla2.AVI et AZERT.AVI fournis avec le magazine.

Basic11 (Société Controlord) : pour l'assemblage et le chargement du programme. La nouvelle version (n°6) se voit dotée d'un simulateur ! On peut en essayer une version limitée disponible en téléchargement sur le site controlord.fr