

## **INTERFACE UNIVERSELLE POUR TO7 ET TO7-70 ET MO5**

**Jean LASCOURS**

### **PRÉSENTATION DE LA CARTE IU07**

La carte IU07 est logée dans un boîtier d'extension identique à celui d'un contrôleur de communication qui se place de la même façon à l'arrière d'un TO7.

Un câble plat de 26 conducteurs, terminé par un connecteur, peut être branché sur différentes cartes, suivant l'application mise en œuvre.

La sortie de cette carte comporte en particulier 16 fils que l'on peut programmer en entrée ou en sortie (PA et PB) et les 4 fils CA1, CA2, CB1, CB2.

Par programme Basic, LSE, Assembleur... la carte IU07 pourra piloter les modules suivants :

- Détection de fermeture ou d'ouverture de contacts (interrupteurs manuels, détecteurs magnétiques, détecteurs de butée...).
- Commande de relais : la carte peut comporter 16 relais indépendants.

L'alimentation des relais (12 V) peut être située sur la carte et les relais peuvent commuter des circuits basse tension ou des circuits sur le réseau 220 V.

On peut donc commander des ampoules, des électro-aimants, des sirènes, des moteurs, des télécommandes diverses.

- Une carte pourra supporter 8 relais et 8 entrées pour détection de contacts.
- Programmation de mémoires EPROM

Divers modules permettent la programmation des 2716, 2732, 2532 d'une part et 27128-27256 d'autre part.

Si cela devait répondre à un besoin une carte pour 2708 pourrait être construite.

Les programmes en Assembleur, associés permettent différentes opérations :

- Test de virginité
- Copie d'une EPROM en mémoire
- Mise en fichier binaire sur cassette ou disque
- Tests de comparaisons
- Programmation, duplication.
- Programmation de mémoires PROM à fusible

Des opérations de décodage d'adresse peuvent être facilitées par l'utilisation de PROMS du type 7611 ou autres, de nombreuses applications peuvent faire appel à ce type de mémoires.

Toutes les opérations possibles avec les EPROM peuvent être envisagées.

### **. Connexion à des appareils de mesure**

Munis de leurs interfaces nécessaires, intégrées ou non, de nombreux appareils de mesure peuvent communiquer avec un micro-ordinateur, la carte IUO7 peut permettre une mise en œuvre facile.

### **. Connexion à du matériel d'expériences**

Les travaux pratique de Physique conduisent souvent à des saisies de position d'objets. Des capteurs optiques reliés à la carte permettent une saisie directe ; la programmation en assembleur permet la saisie de phénomènes rapides.

Associée à des convertisseurs Analogiques-Digitaux, la carte permet d'effectuer des mesures diverses : vitesse, tensions, températures...

De très nombreuses applications pourront être envisagées par la suite ; je suis persuadé que l'imagination des lecteurs qui apporteront leurs suggestions, permettra d'allonger rapidement cette liste.

## PROGRAMMATION DE L'INTERFACE

Pour ceux qui ne connaissent pas l'assembleur (il n'y a pas de honte à ça, on peut très bien vivre sans), précisons que beaucoup d'applications pourront être mises en œuvre sous Basic, LSE... , la programmation est très simple et divers exemples seront donnés.

Pour des applications nécessitant une grande vitesse d'exécution, des programmes assembleur seront disponibles, tout faits. La littérature abonde en articles traitant de la programmation des PIA, cette théorie ne sera pas abordée dans cet article. Nous évoquerons simplement quelques idées de base, élémentaires, permettant de programmer le 6821.

### LE 6821 D'UN COUP D'OEIL

Cette petite merveille est un circuit intégré de 40 pattes dont 20 broches sont tournées vers le micro-ordinateur et 20 broches tournées vers l'extérieur.

Côté micro :

- 8 fils de données pouvant transiter dans les deux sens
- 4 fils de données de contrôle
- 4 fils permettant d'adresser les registres internes 4 fils pour l'alimentation, et signaux divers.

Côté extérieur

- 16 fils répartis en deux "ports" de 8 bits nommés A et B.
- 4 fils ayant un rôle particulier (interruptions...).

### L'INITIALISATION DU PIA

Etude sommaire :

Pour chaque port (A et B), il faut connaître l'existence de deux registres internes (leur adresse est donnée plus loin) :

- le premier est appelé DDRA (ou DDRB pour l'autre port)
- le contenu de ce registre "explique" au PIA dans quel sens vont cheminer les informations.

Il est accessible à l'utilisateur ; les 8 bits correspondent aux 8 fils.

Un 0 indique que le fil est en entrée, un 1 indique une sortie.

Pour mettre tous les fils en entrée sur un port, il suffit donc d'écrire 0 dans le DDR.

Pour mettre tous les fils en sortie on écrit FF soit 1111 1111 en binaire dans le DDR.

Si on veut un fil sur deux en entrée et un fil sur deux en sortie on écrit par exemple '55 dans le DDR ('55 = 0101 0101) ; ici le bit de poids faible est en sortie, pour avoir l'inverse on écrit 'AA dans le DDR ('AA = 1010 1010).

. le second registre est appelé PRA (PRB pour l'autre port). Il contient tout simplement les données lues sur les fils programmés en entrée, ou les données à écrire sur les fils programmés en sortie.

Pour programmer le PIA il suffit donc :

- d'écrire dans le DDR pour préciser le sens des échanges
- de prélever le contenu du PR si on est en entrée et d'écrire dans le PR pour une sortie.

Habituellement, pour une application donnée l'initialisation consiste à préciser les sens de transfert d'information en début, puis à sélectionner le registre PR qui sera sollicité autant de fois que nécessaire, pour les échanges.

## **COMMENT ADRESSER LE DDR ET LE PR ?**

Sur TO7 L'adresse du DDRA est : E7CC

L'adresse du PRA est : E7CC

Vous avez bien lu et il n'y a pas d'erreur de frappe, l'adresse est la même !

Les deux registres sont distincts à l'intérieur du PIA, un aiguillage interne permet de sélectionner l'un ou l'autre. Pour commander cet aiguillage l'utilisateur a accès à un troisième registre appelé CRA qui est un registre de contrôle.

C'est plus précisément le bit 2 qui permet cette sélection. Le CRA de la carte IU07 est à l'adresse E7CD.

Si ce bit est mis à 0, le registre DDRA est accessible à l'adresse E7CC.

Si ce bit est à 1, le registre PRA est accessible à la même adresse : E7CC.

### En résumé

L'initialisation, en pratique revient à :

- écrire 0 à l'adresse E7CD ceci met le bit 2 de CRA à 0 ce qui sélectionne le DDRA en E7CC
- écrire à l'adresse E7CC l'octet image de l'état de chaque fil (entrée = 0, sortie = 1) pour le choix des entrées ou sorties.
- écrire 04 (00000100) à l'adresse E7CD la mise à 1 du bit 2 donne accès au PRA pour échanger les données, par la suite.

C'est tout, il suffit de lire le contenu de E7CC pour une entrée ou d'y écrire les données à sortir.

Cette description, volontairement simplifiée à l'intention des débutants, passe sous silence de nombreuses autres possibilités du PIA mais peut dans un premier temps couvrir une part importante des besoins.

Note : DDR(A) = DATA DIRECTION REGISTER = REGISTRE DE DIRECTIONS

PR(A) = PERIFERAL REGISTER = REGISTRE DE DONNEES

CR(A) = CONTROL REGISTER = REGISTRE DE CONTROLE

LES ADRESSE DE IU07

T07 E7CC : DDRA et PRA : A7CC

E7CD : CRA : A7CD

E7CE : DDRB et PRB : A7CE

E7CF : CRB : A7CF

## QUELQUES EXEMPLES POUR DEBUTER

### Allumer une ampoule

. On utilise la carte équipée de 8 entrées contacts et 8 sorties relais.

. Les 8 relais sont connectés au port A. Ils sont activés s'ils reçoivent un 1.

. On veut allumer la première ampoule et laisser les autres éteintes.

Il faut écrire un 1 sur le fil de sortie correspondant à ce relais et des 0 sur les autres fils.

Basic	Assembleur	Commentaires
10 DORA = &HE7CC	DDRA EQU \$E7CC	Initialisation du PIA Définition des constantes
20 PRA = DDRA	PRA EQU DORA	
30 CRA = PRA + 1	CRA EQU PRA + 1	
40 A = 1 : B = 4	LDD # \$0104	D est la concaténation de A et B
50 POKE CRA, 0	CLR CRA	Accès au DDRA(bit2 à 0)
60 POKE DDRA, A	STA DDRA	1er fil en sortie(fil PA0)
70 POKE CRA, B	STB CRA	Accès au PRA (BIT 2 à 1) Initialisation terminée
80 POKE PRA, A	DEB STA PRA	DEB est une étiquette utile pour la suite du programme Le fil PA0 reçoit 1 : il active le relais. L'ampoule s'allume.

L'activation du relais peut également servir à commander un moteur, une sirène...

### Faire clignoter cette ampoule

. Le programme précédent a permis de l'allumer, il faut attendre quelques instants, éteindre l'ampoule, attendre à nouveau et recommencer le cycle.

Programme à rajouter à la suite :

L'ampoule est allumée

100 GOSUB 1000	BSR TEMPO	Appel du S.P. d'attente
110 POKE PRA, 0	CLR PRA	Fil PA0 mis à 0 La lampe est éteinte
120 GOSUB 1000	BSR TEMPO	Nouvelle attente
130 GO TO 80	BRA DEB	On boucle
1000 FOR 1=1 TO 500	TEMPO LDX#\$8000	S.P. d'attente
1010 NEXT 1	BOUC LEAX,-X	
1020 RETURN	BNE BOUC	
	RTS	

L'ampoule clignote, les moyens de l'arrêter sont : CNT C, initialisation programme ou encore couper le courant!

Le lecteur pourra à titre d'exercices faire clignoter une autre ampoule, plusieurs à la fois, réaliser un clignotement alterné, modifier la fréquence de clignotement... c'est amusant.

## TESTER L'ÉTAT D'UN INTERRUPTEUR

- . On utilise la carte équipée de 8 entrées contacts et B sorties relais.
- . Les 8 entrées contacts sont connectées au port B
- . Les interrupteurs placés entre l'entrée et la masse imposent un 0 s'ils sont fermés (ils laissent passer le courant) et un 1 s'ils sont ouverts (position OFF ou ARRET).

Basic	Assembleur	Commentaires
		Initialisation PIA
DDRB=&HE7CE	DDRB EQU \$E7CE	Définition des constantes
PRB=DDRB	PRB EQU DDRB	
CRB=PRB + 1	CRB EQU PRB + 1	
A=0 : B=4	LDD # \$0004	
POKE CRB,A	STA CRB	Accès au DDRB
POKE DDRB,A	STA DDRB	Tous les fils en entrée
POKE CRB,B	STB CRB	Accès au PRB
		Initialisation terminée

Il suffit d'examiner le contenu de PRB pour connaître l'état des huit interrupteurs.

- en Basic : X=PEEK (PRB) suivi du traitement (examen, affichage...)
- en assembleur : LDA PRB

Le registre A contient alors l'image des huit états. Il suffit d'effectuer les comparaisons et d'exécuter la suite du traitement.

A titre d'exercice on pourra installer un poussoir sur l'un des fils du port B, brancher une ampoule sur un relais du port A et démarrer le clignotement dès que l'on a appuyé puis l'interrompre au nouvel appui.

A suivre...

Jean LASCOURS  
Collège Lakanal  
09000 FOIX

```

**** Definition des constantes ****
E7CC DDRA EQU $E7CC
E7CC PRA EQU DDRA
E7CD CRA EQU PRA+1

**** Programme ****
A000 ORG $A000
A000 CC 0204 LDD ##0204 A=1(PA0) B=4
A003 7F E7CD CLR CRA Acces au DDRA
A006 B7 E7CC STA DDRA PA0 en sortie
A009 F7 E7CD STB CRA Acces au PRA
A00C 86 00 DEB LDA #$0 Lampe eteinte
A00E B7 E7CC STA PRA
A011 8D 09 BSR TEMPO
A013 86 02 LDA #$2 Lampe allumee
A015 B7 E7CC STA PRA
A018 8D 02 BSR TEMPO
A01A 20 F0 BRA DEB On reboucle

**** Sous-programme temporisation ****

A01C 8E 8000 TEMPO LDX ##8000
A01F 30 82 BOUC LEAX , -X
A021 26 FC BNE BOUC
A023 39 RTS

0000 END

00000 Total Errors

1 **** CLIGNOITEUR ****
2 'J.Lascours 1/85
10 **** Initialisations ****
20 CRA=&HE7CD:DDRA=CRA-1:PRA=DDRA
50 POKE CRA,0 'Acces au DDRA
40 POKE DDRA,2 'Bit 1 en sortie
50 POKE CRA,4 'Acces au port E/S
60 T=200 'vitesse clignotement
70 '
80 **** Programme ****
90 IF (PEEK(PRA)AND &B00000100)=&B100 THEN 90 'Test de l'
interrupteur
100 POKE PRA,2 ' Bit 2 active a 1
110 FOR J=1 TO T:NEXT J ' Tempo
120 POKE PRA,0 ' Bit 2 remis a zero
130 FOR J=1 TO T:NEXT J ' Tempo
140 GOTO 90
ASO' ICI L'AMPOULE CLIGNOTE TANT QUE L'INTER EST FERME

```

#### DESCRIPTION DE L'INTERFACE IU07

L'électronique se trouve réduite à sa plus simple expression : un connecteur 2x19 points adapté au T07  
un connecteur 2x13 points adapté au câble plat de sortie

un circuit imprimé simple face  
un PIA 6821  
un condensateur 10µ F 25 V goutte tantale  
c'est tout !

Les habitués de la bidouille seront quelque peu surpris (agréablement cependant) de l'absence de circuits décodeurs d'adresse.

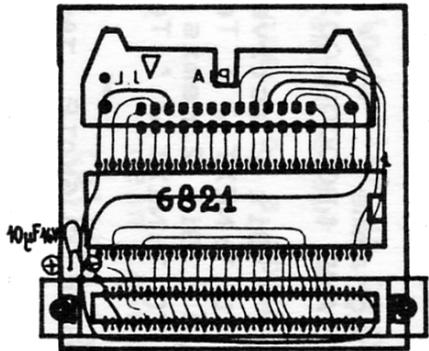
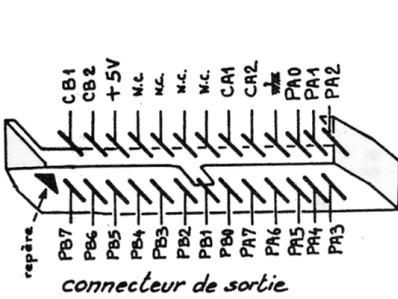
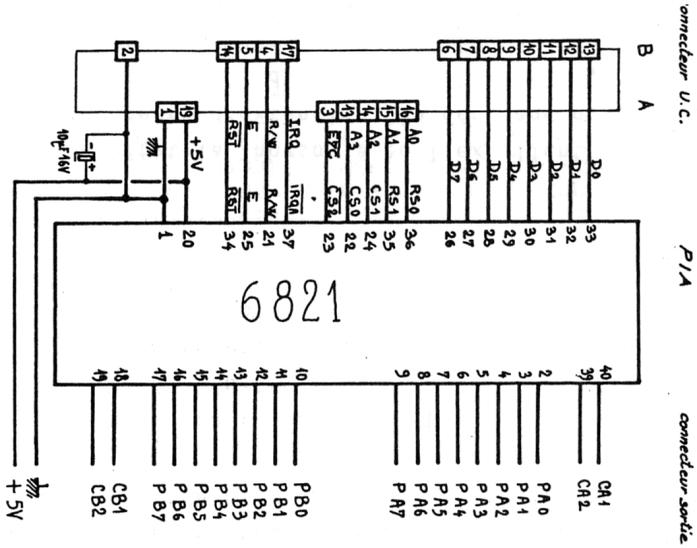
En effet la Thomson a eu l'excellente idée de sortir sur les connecteurs d'extension, le décodage du bloc E7C.

L'interface, adressée en E7CC à E7CF, ne nécessite donc aucun circuit de décodage.

Ces adresses correspondent à celles de l'interface manettes de jeux.

Il faut donc ôter cette dernière pour utiliser IU07, et inversement.

CARTE IU 07



J. Lascours