Acquisition numérique automatisée d'une tension

Analyse d'une configuration matérielle et logicielle

par P. FRETAUD, J.-P. HUMEAU et B. VELAY
Département Mesures Physiques, IUT de Saint-Nazaire

INTRODUCTION

Pour expliciter le fonctionnement des systèmes de type ORPHY ou CANDIBUS, ou encore celui des oscillo à mémoire numérique, nous analysons un **module d'acquisition de tension** en présentant trois points de vue complémentaires : le **schéma électronique**, les **chronogrammes** des principaux signaux de commande mis en jeu, le **logiciel didactique** (en Turbo Pascal) qui le commande.

Cet article, issu d'une manipulation d'électronique pratiquée en 2ème année d'IUT Mesures Physiques, présente le **processus de conversion analogique/numérique d'une tension** puis décrit en détail l'**interfaçage de la carte électronique avec l'ordinateur PC qui la commande**. Il insiste ainsi sur les **deux emplois des signaux numériques** manipulés, à savoir **coder des données numériques** (par exemple les nombres issus des conversions) ou **coder des groupes de commande.**

PRINCIPE DE LA MÉTHODE EMPLOYÉE

Rappelons celui de la mesure d'une tension avec un voltmètre à aiguille classique : après avoir choisi les points de branchements des fils et le calibre, on suit le déplacement de l'aiguille sur une graduation afin de déterminer celle qui convient ; on en tire par interpolation, la valeur «lue», exprimée avec un nombre fini de chiffres. La **«conversion»** analogique (la tension d'entrée) / numérique (le nombre retenu pour la lecture) est donc **faite «à vue» !**

Bien que cette comparaison ait des limites (en particulier pour l'analyse des erreurs), elle montre qu'**il n'y a rien de fondamentalement nouveau** si ce n'est l'automatisation !

La tension à mesurer est choisie parmi les tensions disponibles sur les différentes entrées de la carte. Cette tension pouvant varier au cours

du temps, il faut donc choisir la valeur particulière qui va être mesurée : c'est la valeur de la tension au moment de l'échantillonnage qui est figée («bloquée») puis convertie en un nombre binaire qui est lu par l'ordinateur. La valeur de la tension est calculée en tenant compte du

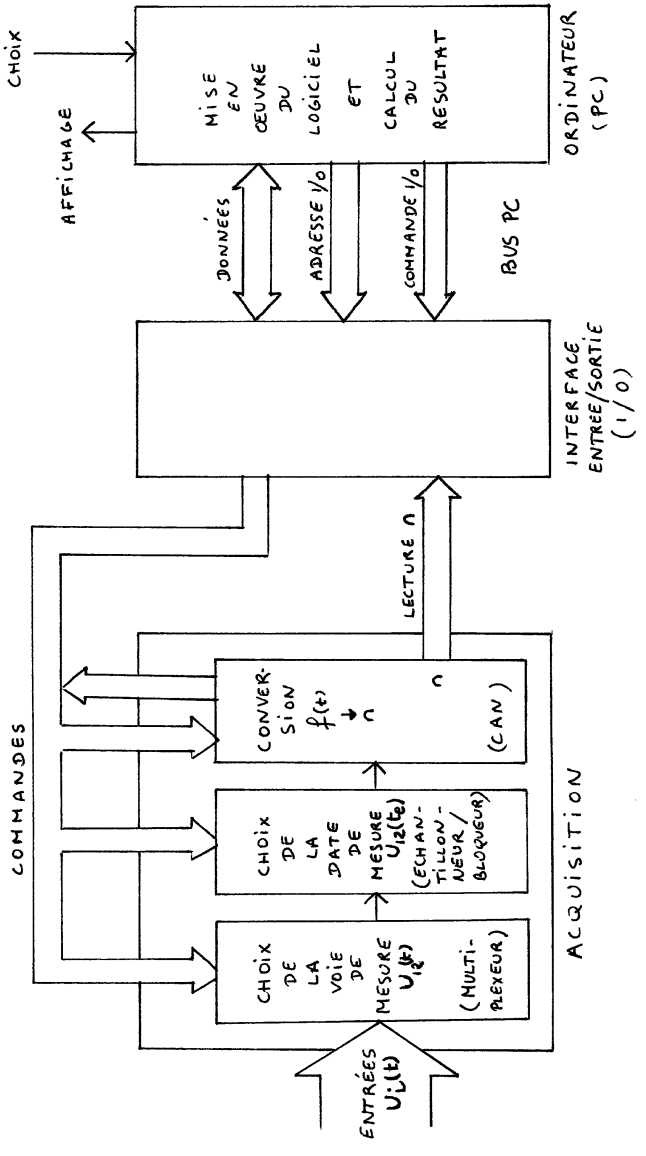

Figure 1 : Synoptique du système complet.

calibre, du nombre total de «graduations» et du nombre lu (la conversion ayant lieu sur 12 bits, il y a $2^{12} = 4096$ valeurs différentes pour le résultat de la lecture). **L'ordinateur joue ici le rôle de l'opérateur** qui enchaîne dans l'ordre (du logiciel) les commandes adéquates, les lectures, les calculs et enfin les affichages des résultats sur l'écran ou l'imprimante.

ARCHITECTURE DE LA CARTE D'ACQUISITION

Rappel : un «bit» correspond à un chiffre dans la base 2 (0 ou 1) ; par exemple un nombre codé sur 8 bits est donc composé de 8 «chiffres» binaires, soit une succession de huit 0 ou 1. Les nombres sont compris entre 0 et $2^8 - 1 = 255$.

Les fonctions de commande qui vont être décrites par la suite, sont à **deux états**. Ces deux états sont donc codables par 0 ou 1 : **une fonction peut donc correspondre à un «bit»**. En particulier lors de l'étude des chronogrammes, on remarquera le **système de notation** indiqué sur ces exemples :

– CE est une fonction valide à 1, non valide à 0,

– $\overline{\text{CS}}$ est une fonction valide à 0, non valide à 1 (c'est la signification de la barre au dessus),

– R/$\overline{\text{C}}$ est une fonction où R est valide à 1 et C valide à 0 (indication de la barre).

L'examen du schéma de la carte d'acquisition (figure 2) montre en effet :

• **La partie «choix et conditionnement du signal analogique»**, réalisée par le circuit Analog Device 364 AIS (analog input section) :

– les **entrées analogiques** reliées aux différentes tensions mesurables ① : la carte peut ainsi suivre successivement jusqu'à 16 tensions différentes référencées à la même masse, ou jusqu'à 8 tensions non référencées à la même masse (**mode «single ended» ou «différentiel»**) ;

– le **multiplexeur analogique** ② (il joue le rôle de la «gare de triage» pour le réseau) dont on choisi la voie scrutée pour la mesure par les commandes ③. Les **verrous** (latches) servent à stocker l'**adresse** de la voie choisie (exprimée sur 4 bit $2^4 = 16$) pendant toute la **durée de la mesure**. Nous précisons plus tard les autres commandes ;

– l'**échantillonneur-bloqueur** ④ (sample and hold $\overline{\text{S}}$/H) **fige** le signal à convertir pendant toute la durée de la mesure : l'échantillonnage peut être périodique si la commande est reliée à une horloge ; **la fréquence**

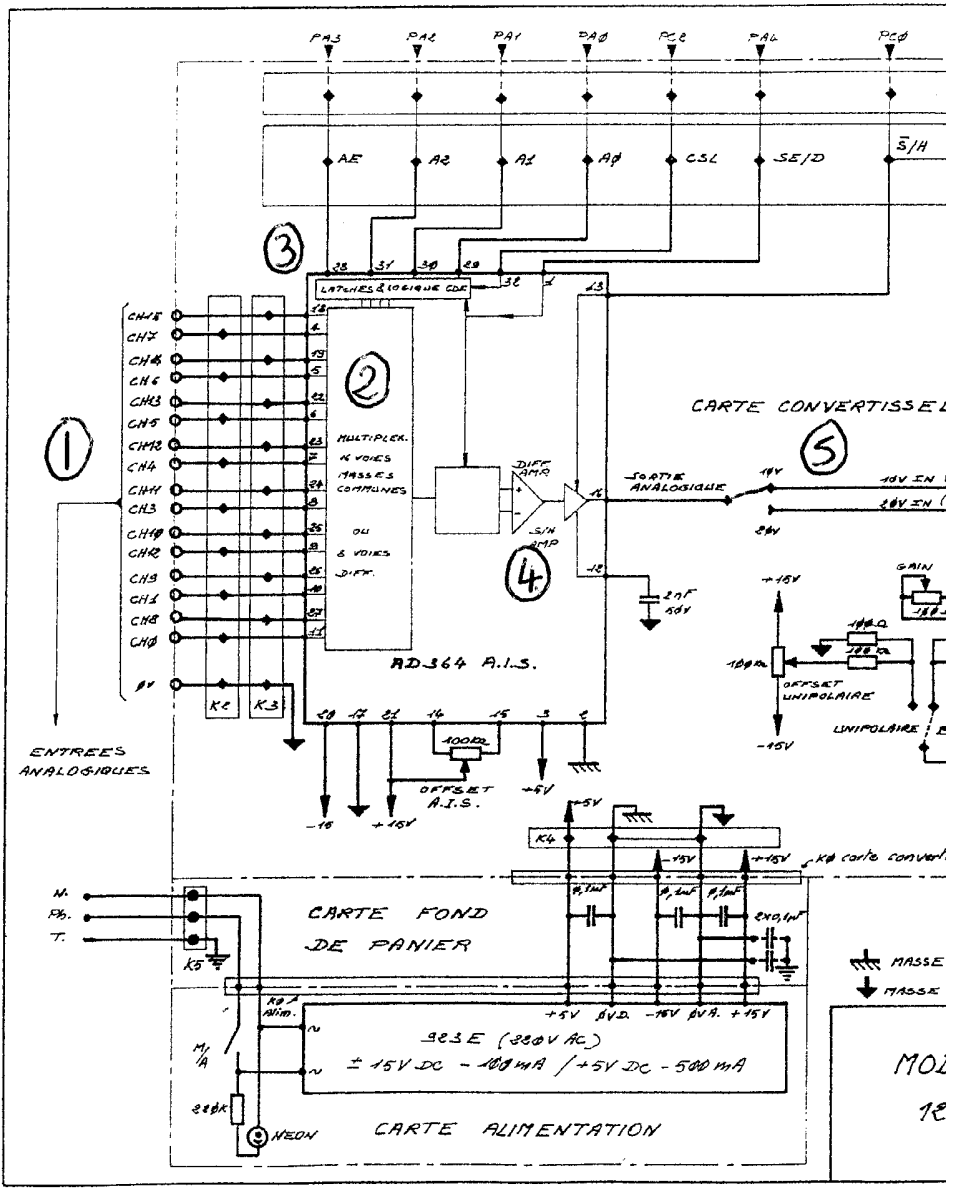
PA3
PA2
PA1
PA0
PC2
PA4
PC0
AE
A2
A1
A0
CSL
SE/D
S/H
3
LATCHES & LOGIQUE CDE
CH15
CH7
CH14
CH6
CH13
CH5
CH12
CH4
CH11
CH3
CH10
CH2
CH9
CH1
CH8
CH0
0V
2
MULTIPLEX.
16 VOIES
MASSES
COMMUNES
OU
8 VOIES
DIFF.
DIFF AMP
S/H AMP
1
4
5
CARTE CONVERTISSEUR
SORTIE ANALOGIQUE
+15V
-15V
2nF
50V
AD364 A.I.S.
K2
K3
ENTREES ANALOGIQUES
100Ω
OFFSET A.I.S.
+5V
K4
CARTE FOND DE PANIER
N.
Ph.
T.
K5
K0 Alim.
9283E (220V AC)
± 15V DC - 100mA / +5V DC - 500mA
220K
NEON
CARTE ALIMENTATION
MASSE
GAIN
OFFSET UNIPOLAIRE
UNIPOLAIRE

Figure 2

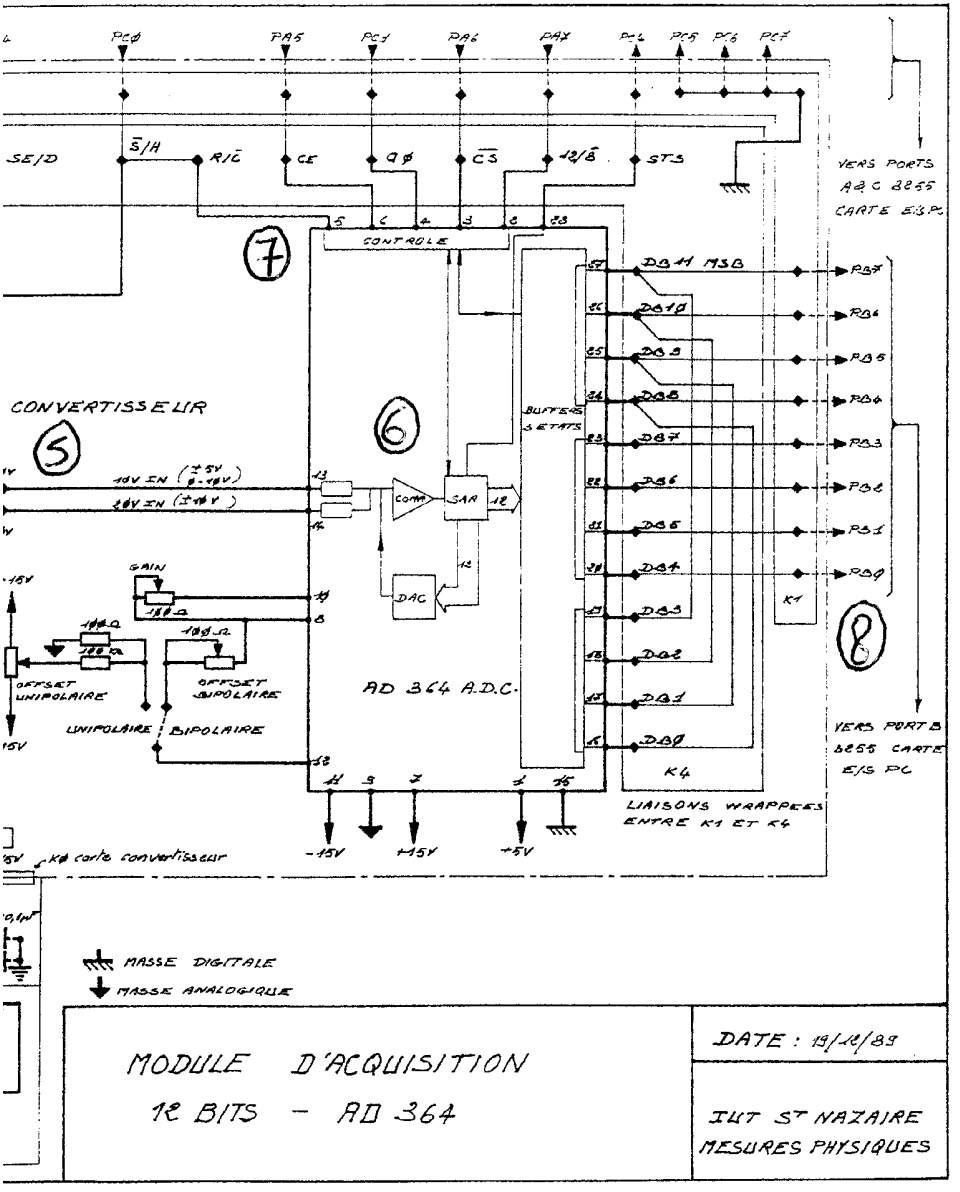
PC0
PA5
PC1
PA6
PA7
PC4
PC5
PC6
PC7
SE/D
S/H
R/C
CE
Q0
CS
12/8
STS
VERS PORTS A,B,C 8255 CARTE E/S PC
CONTROLE
7
DB11 MSB
DB10
DB9
DB8
DB7
DB6
DB5
DB4
DB3
DB2
DB1
DB0
PB7
PB6
PB5
PB4
PB3
PB2
PB1
PB0
CONVERTISSEUR
5
6
BUFFERS 3 ETATS
10V IN (±5V 0-10V)
20V IN (±10V)
COMP
SAR
12
DAC
GAIN
100Ω
100Ω
100Ω
OFFSET UNIPOLAIRE
OFFSET BIPOLAIRE
UNIPOLAIRE
BIPOLAIRE
AD 364 A.D.C.
K1
K4
8
VERS PORT B 8255 CARTE E/S PC
LIAISONS WRAPPEES ENTRE K1 ET K4
-15V
+15V
+5V
K0 carte convertisseur
MASSE DIGITALE
MASSE ANALOGIQUE
MODULE D'ACQUISITION
12 BITS - AD 364
DATE : 19/12/89
IUT ST NAZAIRE
MESURES PHYSIQUES

Figure 2

d'échantillonnage est limitée par la durée du cycle complet conversion/lecture ; ce sont des grandeurs-clé puisque, par exemple, elles limitent en fréquence les signaux étudiables ;

– la **calibration** du signal analogique prêt à être converti ⑤.

• **La partie «conversion/lecture»** réalisée par le circuit AD364 ADC (analog/digital converter)

– le **convertisseur analogique/numérique ⑥ est un convertisseur par approximations successives** (SAR : successive approximations registers) élabore le nombre binaire correspondant (voir Annexe 1) du **digit binaire de poids le plus élevé** 2^{11} (MSB most significant bit) au **digit binaire de poids le moins élevé** 2^0 (LSB less significant bit) ;

– les **signaux de commande** sont disponibles ⑦ et déclenchent les **étapes successives**. Ils sont gérés en fait par le logiciel et sont transmis par une interface ;

– le **nombre issu de la conversion** est **disponible en 12 bits** sur les 12 broches DB0 à DB11 ⑧, mais il y a deux possibilités de lecture : soit les données sont lues en une volée (mode 12 bits : $12/\overline{8} = 1$), soit les données ne peuvent être **transmises** qu'en **8 bits** (le port B n'a que 8 lignes) et la lecture doit se faire en **deux volées successives de 8 bits chacune** (**mode 8 bits**: $12/\overline{8} = 0$) tout d'abord les 8 bits de poids les plus forts (MSB) puis les 4 bits de poids les moins forts (LSB) ainsi que 4 bits redondants qui seront éliminés par la suite.

La **reconstitution du résultat de lecture complet** (en 12 bits) et sa **traduction en tension mesurée** sont effectuées par le logiciel.

ARCHITECTURE DE L'INTERFACE ORDINATEUR PC / CARTE D'ACQUISITION

Pour que le logiciel décrit plus bas puisse «prendre en main» la mesure de tension, il faut que **l'ordinateur PC communique avec le système d'acquisition** par l'intermédiaire d'une carte d'entrée/sortie PPI 8255 (Programmable Peripheral Interface) de Intel. On appelle aussi ce systèe PIO (Parallele / Input Output / **Entrée Sortie par communication parallèle**).

La figure 3 montre les différentes liaisons entre les Bus de l'ordinateur, la carte d'entrée/sortie et la carte d'acquisition de données.

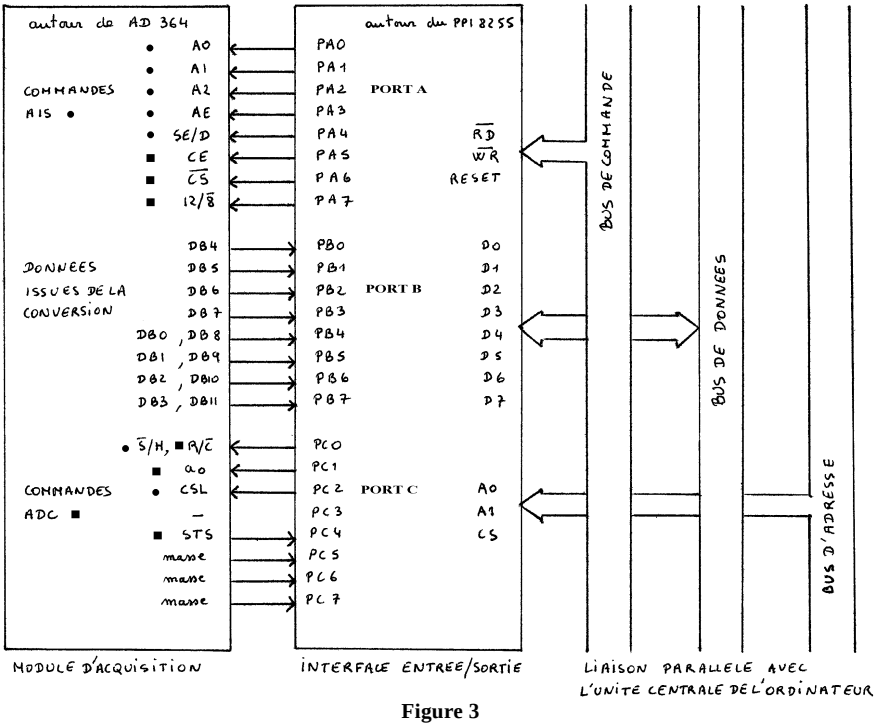

Figure 3

L'interface communique avec la carte d'acquisition par trois registres (ou **«ports»**) A, B et C **de huit bits chacun**. L'affectation en entrée ou en sortie des lignes reliées aux registres se fait en **configurant la carte** (voir programme) selon divers modes. Le choix se fait donc selon les besoins : par exemple, ce module nécessite une réponse de la carte vers l'ordinateur («status» indique à l'ordinateur la fin de conversion), il faudra configurer le port C moitié en entrée, moitié en sortie.

Les huit broches D0 à D7 constituent l'extrémité du **«bus de données»** (groupe de fils assurant une liaison parallèle avec l'ordinateur) **qui permet de faire transiter, sous forme de «mots» binaires de 8 bits des données ou des commandes (vers les ports A B C ou en provenance de ceux ci**) ou encore **d'obtenir la configuration au moyen d'un mot de contrôle**.

Il est très important de comprendre que les «mots» binaires qui transitent sur les ports et qui correspondent à des nombres (exprimés

dans le programme soit en **binaire** soit en **hexadécimal**) peuvent contenir soit des données, soit des ordres de commandes. Ces données (data) issues par exemple de la conversion (port B) ont donc un sens en tant que nombres et peuvent être exploitées (calcul de la valeur). Ces commandes, par exemple le choix de la configuration ou bien encore les ordres de conversion ou de lecture (port A ou C), changent à chaque étape du processus. **Écrire le logiciel a donc pour but de déterminer les étapes puis de coder les ordres nécessaires** sous forme de suite de 0 et de 1 regroupés **par mots de 8 bits**, on parle de **codage sur 8 bits**.

Le **«bus d'adresse»** est constitué de plusieurs lignes sur lesquelles transitent des signaux qui élaborent la commande **CS** («chip select» circuit sélectionné) autorisant la communication avec l'unité centrale de l'ordinateur et des lignes $\mathbf{A_0}$ et $\mathbf{A_1}$ (codage sur 2 bits $2^2 = 4$) qui permettent **à l'ordinateur de choisir l'une des quatre adresses de destination (en hexadécimal)**.

Port A : 300H Port B : 301H Port C : 302H Contrôle : 303H

L'instruction **PORT [$xxx]:=$xx** de Turbo Pascal permet à l'ordinateur d'écrire sur le port concerné le mot binaire requit via le bus de données, l'instruction **VARIABLE := PORT[$xxx]** permet à l'ordinateur de lire sur le port concerné et d'affecter la valeur lue à la variable indiquée (voir le programme et l'Annexe 2 pour les détails).

Le **«bus de commande»** est constitué de lignes supplémentaires sur lesquelles transitent trois commandes validant les liaisons avec l'ordinateur :

– $\overline{\text{RD}}$ («read» lire) autorise l'unité centrale à lire les informations présentes sur les «bus»,

– $\overline{\text{WR}}$ («write» écrire) autorise l'unité centrale à écrire les mots de contrôle ou de commande sur les «bus» ,

– RESET («remise à zéro» ou « réinitialisation»).

L'ensemble de ces trois groupes de fils constitue au sens strict **le** «bus» de liaison avec le PC.

CHRONOGRAMMES DES SIGNAUX DE COMMANDE DE LA CARTE D'ACQUISITION PENDANT UN CYCLE DE MESURE

Tout d'abord examinons les **commandes liées à la section «entrée analogique»** (AIS).

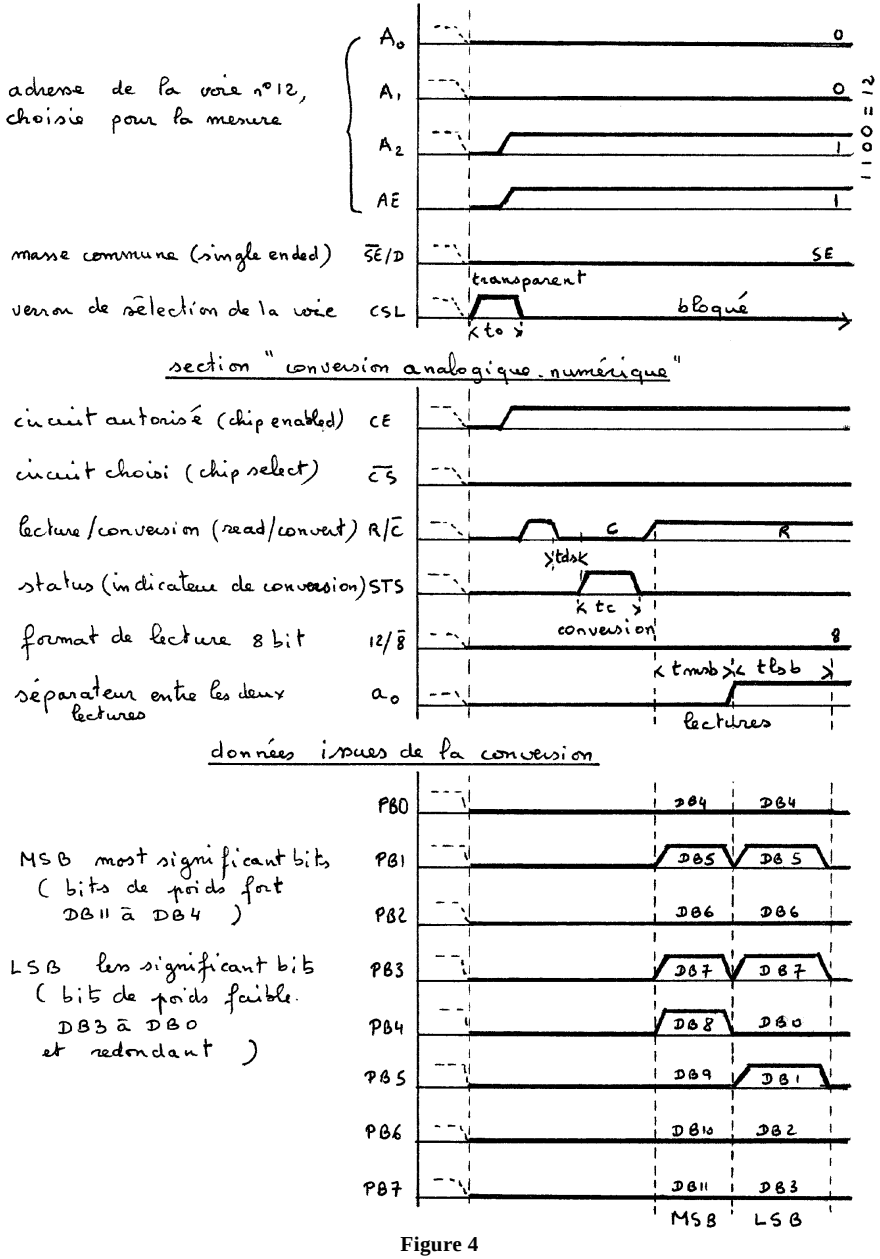
adresse de la voie n°12, choisie pour la mesure
A0
A1
A2
AE
0
0
1
1
1100 = 12
masse commune (single ended)
SE/D
SE
transparent
verrou de sélection de la voie
CSL
bloqué
tco
section "conversion analogique-numérique"
circuit autorisé (chip enabled)
CE
circuit choisi (chip select)
CS
lecture/conversion (read/convert)
R/C
C
R
tdsk
status (indicateur de conversion)
STS
tc
conversion
format de lecture 8 bit
12/8
8
séparateur entre les deux lectures
a0
tmsb
tlsb
lectures
données issues de la conversion
MSB most significant bits (bits de poids fort DB11 à DB4)
LSB less significant bit (bits de poids faible DB3 à DB0 et redondant)
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
DB4
DB4
DB5
DB5
DB6
DB6
DB7
DB7
DB8
DB0
DB9
DB1
DB10
DB2
DB11
DB3
MSB
LSB

Figure 4

Le **numéro de voie choisi** pour l'exemple est 12.

En décomposant 12 en binaire, il vient : $12 = 2^3 + 2^2 = 1\ 100$ **codé sur quatre bits**

L'adresse à coder est donc : AE = 1 $A_2 = 1$ $A_1 = 0$ $A_0 = 0$

D'autre part le choix du numéro 12 implique le fonctionnement en mode «masse commune» (single ended), donc $\overline{SE}$ /D = 0.

Le choix de la voie est fait pendant le durée to en début de cycle en modifiant les registres A_0, A_1, A_2 et AE lorsque le **verrou est transparent** (CSL = 1 Channel Select Latch verrou de sélection de la voie). Enfin, pendant la conversion et la lecture, **le contenu de ces registres d'adresse est verrouillé** afin de fixer la voie scrutée (CSL = 0).

Ensuite, examinons les **commandes liées à la section «conversion analogique-numérique »**(ADC).

Les signaux correspondent à l'exemple détaillé en annexe 1.

D'abord le **circuit de conversion est validé** (CE = 1 Chip Enabled circuit autorisé) puis **choisi** (CS = 0 Chip Select).

L'entrée de commande de l'échantillonneur-bloqueur ($\overline{S}$/H) et la commande de lecture et conversion (R/$\overline{C}$) du convertisseur sont reliées (PC0 cf figure 2). Lorsque l'on applique un «0» logique sur la ligne PC0, simultanément l'échantillonneur-bloqueur se bloque et la conversion commence. **Le début de la conversion** est donc lié au **passage à 0** de R/$\overline{C}$. Après un retard maximum de 500 ns (tds time delay status) la conversion commence (STS, status, passe à 1). La durée maximum de conversion (tc) est de 35 µs. **La fin de conversion est indiquée à l'ordinateur par le passage à 0 de la ligne «status »** c'est cet événement qui est testé (et attendu) par le programme.

Enfin le format sous lequel seront lues les données est choisi (12/$\overline{8}$ = 0, lecture de données en 8 bits). En effet (voir annexe 1) le convertisseur peut travailler en 12 ou en 8 bits ; dans ce second cas, **il faut lire les bits issus de la conversion en deux fois** Pendant la lecture des bits de poids forts MSB la commande **a0** est à 0 (tmsb = 16 µs maxi) puis a_0 passe à 1 pour indiquer qu'il faut lire les bits de poids faibles LSB (tlsb = 9 µs maxi).

PROGRAMME DE COMMANDE DU MODULE D'ACQUISITION

```
PROGRAM CAN12bit;  { CONVERTISSEUR 12 BITS AD 364 }

{ Programme Pascal commandant la carte E/S PC implantée dans l'ordinateur : }
{ Les instructions données au convertisseur et la lecture des mesures }
{ se font par l'intermédiaire de mots binaires utilisants 8 bits ( octets ) }

USES crt;
VAR lsb,msb,valeur,numero : integer;
        tension : real;
        choix  : char;
        SD : integer;          { SE/DIFF }
        CE : integer;          { Chip Enabled }

BEGIN
clrscr;

{ Choix de la voie de mesure de tension et du mode }

writeln('  CHOIX DU MODE D'ENTREE ');
writeln('S - Entrées masses communes');
writeln('D - Entrées différentielles');
writeln;
repeat
      write('Votre choix ?  : ');
      readln(choix);
until choix in ['S','D'];
writeln;
case choix of  'S' : SD := 0 ;
               'D' : SD := 1 ;
               end;

writeln('  CHOIX DU NUMERO DE LA VOIE D'ENTREE ');
writeln('0 ... 15 - Entrées masses communes');
writeln('8 ... 15 - Entrées différentielles');
writeln;
write('Numéro de voie ?  : ');
readln(numero);

CE := 1;

{ les adresses des ports sont fixées par un choix de cavaliers sur la carte }
{ port A : $300 , port B : $301 , port C : $302 , port de contrôle : $303 }

{ Mot de contrôle assurant la configuration des ports de la carte d'entrée/sortie E/S PC }
port[$303] := $8A;    { $ précède un nombre noté en hexadécimal de 0 ... F=15 }

(*                          CONFIGURATION DES PORTS                          *)
(*     PORT A              PORT B                  PORT C                   *)
(*  sortie PA0:A0      | entrée PB0:DB4        | sortie PC0:R/C             *)
(*  sortie PA1:A1      | entrée PB1:DB5        | sortie PC1:ao              *)
(*  sortie PA2:A2      | entrée PB2:DB6        | sortie PC2:CSL             *)
(*  sortie PA3:AE      | entrée PB3:DB7        | sortie PC3:(dispo.)        *)
(*  sortie PA4:SE/DIFF | entrée PB4:DB8  & DB0 | entrée PC4:STATUS          *)
(*  sortie PA5:CE      | entrée PB5:DB9  & DB1 | entrée PC5:(masse)         *)
(*  sortie PA6:CS      | entrée PB6:DB10 & DB2 | entrée PC6:(masse)         *)
(*  sortie PA7:12/8    | entrée PB7:DB11 & DB3 | entrée PC7:(masse)         *)
```

```
{ Section ENTREE ANALOGIQUE }

{ Le verrou d'adresse de voie est rendu "transparent" pour permettre d'écrire }
{ et de stocker le numéro de la voie de mesure choisie }
port[$302] := $05;                { R/C = 1 , a0 = 0 , CSL = 1 }
{ Mot placé sur le port A codant le numéro de la voie de mesure, et le mode simple ou differentiel }
{ Il active aussi le composant CE=1 }
port[$300] :=(CE shl 5)+(SD shl 4)+(numero);  { ( ) déclare le nombre sous sa forme binaire }
{ Blocage du verrou d'adresse : CSL mis à 0 }
port[$302] := $01;                {R/C = 1 , a0 = 0 , CSL = 0 }


REPEAT { Boucle répétant le cycle complet de conversion }
    begin

        { Début de conversion : R/C mis à 0 }
        port[$302] := $00;              { R/C = 0 , a0 = 0 , CSL = 0 }
        repeat { Attente de la fin de conversion : Status mis à 0 }
        until port[$302] AND $10 =$00 ;   { on teste status = 0 }

        { Autorisation de lecture : R/C mis à 1 }
        port[$302] := $01;              { R/C = 1 , a0 = 0 , CSL = 0 }
        { Lecture des 8 bits les plus signifiants ( msb ) sur le port B }
        msb := port[$301];              { lecture 8 msb parce que a0=0 }
        { Autorisation de lecture des 8 bits les moins signifiants ( lsb ) }
        port[$302] := $03;      { a0 mis à 1 : R/C = 1 , ao = 1 , CSL = 0 }
        { Lecture des 4 bits les moins sigifiants ( et de 4 autres redondants ) }
        lsb := port[$301];


        { Fabrication du résultat sur 12 bits : lsb décalés à droite de 4 rangs, ce qui le réduit }
        { à 4 bits D0 à D3, msb décalés à gauche de 4 rangs pour donner D4 à D11 }
        valeur := (msb shl 4) + (lsb shr 4);
        { Calcul de la tension sur l'échelle 0 V , 10 V }
        { le "pas de numérisation " est 10/4096=2.5 mV }
        tension := (10*(valeur/4096));
        { Affichage du résultat de la mesure }
        writeln(msb,'  ',lsb,'    ',valeur,'        ',tension:10:3);
      end;

until keypressed;
END.
```

On peut consulter en Annexe 2 le détail des ordres de commande employés par le logiciel au moyen de l'instruction PORT.
Par ailleurs, il faut préciser que les spécialistes préfèrent employer le langage machine (assembleur) pour la commande de ce genre de carte. Ils considèrent comme nettement plus facile et plus «lisible» de procéder ainsi puis d'insérer ces lignes de code dans le programme en langage évolué (turbo pascal par exemple) qui traitera les données.

RÉSULTATS

On peut consulter en Annexe 1 un exemple détaillé de conversion.

Les informations données par les fabricants donnent donc une **durée maximum inférieure à 60 µs pour le cycle complet Conversion / Lecture(s)**. Cela correspond à la **fréquence d'échantillonnage maximale de 17 kHz** mentionnée. **La résolution est d'environ 2,5 mV, en 12 bits, sur le calibre 0 - 10 V** (pleine échelle). Il y a 2^{12} = 4096 résultats différents possibles sur l'échelle.

La référence 3 analyse les différentes familles de convertisseurs : il en résulte que rapidité et précision ne vont pas ensemble. En particulier lorsque le nombre de bits augmente, la précision s'améliore mais la durée du cycle s'allonge, à technologie identique : **on choisit donc le nombre de bits requis en tenant compte de la précision voulue**, du coût des composants et de la mise en œuvre.

Le logiciel, voulu très simple, peut être amélioré : par exemple, on peut lui faire scruter successivement les différentes voies (penser aux multitudes de capteurs que nécessite une unité de production industrielle : température, pression, etc...). On peut aussi lui faire faire des mesures à intervalles de temps précisés (suivi de «routine») ou encore tester le résultat (détection de sécurité par exemple), etc...

BIBLIOGRAPHIE

[1] Data Conversion Products Databook 1989/90 ANALOG DEVICE (AD 364).

[2] Notice INTEL 8255A/8255A-5.

[3] **Électronique des systèmes de mesures** par TRAN TIEN Lang (Masson)
(en particulier pour l'analyse des différentes fonctions : échantillonnage-blocage, conversion, multiplexeur, etc...).

[4] **Systèmes de mesures informatisées** par TRAN TIEN Lang (Masson)
(en particulier pour les problèmes de communication avec un ordinateur et pour les problèmes d'automatisation).

[5] Physique Appliquée - Première F2 et Terminale F2 collection MERAT/MOREAU (Nathan)
(en particulier pour les problèmes liés au binaire et à l'hexadécimal, aux opérations sur les nombres binaires telles que décalage, addition et produit logique).

Annexe 1

EXEMPLE DÉTAILLÉ

Le nombre de bits est 12, il y a donc 2^{12} = 4096 valeurs possibles pour le résultats.
La «calibre» est de 10 volt (on préfère l'expression «pleine échelle, PE, qui traduit mieux l'anglais «full scale» FS).
La plus petit différence mesurable ou **«pas de quantification»** est p = 10/4096 = 2.5 mV.
Supposons que l'on mesure la tension d'une pile-étalon de valeur 1.021 V.

CONVERSION PAR APPROXIMATIONS SUCCESSIVES

Tout d'abord, la tension analogique à mesurer, issue de la section d'entrée, est comparée («comp(arator)») à une tension de référence : celle ci, pour le bit le plus signifiant DB11, est obtenue en divisant la tension calibre par 2 (en fait 2 puissance 12-11), soit 5 V. Comme 1.021 < 5, le bit DB11 = 0.

Ensuite, on élabore la nouvelle tension à mesurer en soustrayant à celle-ci la tension correspondant à DB11 en utilisant le convertisseur numérique-analogique (DAC digital/analog converter).
Soit $U_{mes} = 1.021 - 0 \times 5 = 1.021$ V.

On recommence par DB10 en comparant U_{mes} à $U_{cal} / 2^2$ (10 divisé par 2 puissance 12 - 10), soit 2.500 V.
Comme 1.021 < 2.500, on a DB10 = 0. U_{mes} reste égale à 1.021 V.

On recommence ainsi de suite 12 fois.

n° du bit	11	10	9	8	7	6
U_{ref} en V	5.000	2.500	1.250	0.625	0.312 (5)	0.156 (25)
U_{mes} en V	1.021	1.021	1.021	1.021	1.021 – 0.625 = 0.396	0.396 – 0.156 = 0.0835
Valeur du bit	0	0	0	1	1	0

n° du bit	5	4	3	2	1	0
U_{ref} en V	0.078 (125)	0.039 (063)	0.019 (53)	0.009 (76)	0.004 (88)	0.002 (44)
U_{mes} en V	0.083 (5)	0.083 (5) – 0.078 = 0.005 (375)	0.005 (375)	0.005 (375)	0.005 (375)	0.005 (375) – 0.002 (44) = 0.000 (492)
Valeur du bit	1	0	0	0	1	0

LECTURES

Lors de la première lecture, on lira les bits les plus signifiants soient : 00011010 (DB11 à DB4), lors de la seconde lecture on lira les bits moins signifiants soient 00101010 (DB3 à DB0 puis DB7 à DB4). DB7 à DB4 sont lus deux fois : ils sont redondants, il faudra les éliminer par décalage.

RÉSULTAT EN 12 BITS

Les MSB décalés à gauche de 4 rangs : 000110100000

Les LSB décalés à droite de 4 rangs : 000000000010 (1010 est donc éliminé)

d'où le résultat en 12 bits : 000110100010

Cela donne pour la conversion : $2^8 + 2^7 + 2^5 + 2^1$ soit 418.

La tension mesurée est donc $10 \times (418 / 4096)$ soit 1.021 V à au moins 2.5 mV près.

Annexe 2

SIGNIFICATION DÉTAILLÉE DES INSTRUCTIONS DU PROGRAMME

PORT[$303]:=$8A configuration des ports A B C

L'instruction est une «sortie» pour l'ordinateur, puisqu'il affecte un nombre à une adresse.
$303 est l'adresse dans l'unité centrale de l'ordinateur où il faut porter le nombre $8A afin de coder le «mot de contrôle» décrivant la configuration des entrées/sorties de la carte d'interface.

$\$8A = 8 \times 16^1 + 10 \times 16^0 = 138 = 2^7 + 2^3 + 2_1 + 2_0 = 10001010$
Le mot bianire correspond aux 8 bits D7 à D0.
Le code donné par la notice du fabricant INTEL pour la configuration des ports indique :

*D0 = 0	port C lower output (PC0 à PC3 en sortie pour l'unité centrale)
*D1 = 1	port B input (PB0 à PB7 en entrée pour l'unité centrale)
*D2 = 0	mode 0 sélectionné (les lignes sont en permanence soit en entrée soit en sortie)
*D3 = 1	port C upper input (PC4 à PC7 en entrée ; le port C est en fonctionnement mixte)
*D4 = 0	port A output (PA0 à A7 en sortie)
*D5 = 0 et *D6 = 0	mode 0 sélectionné
*D7 = 1	mode set flag (en fait toujours à 1 quelque soit le mode)

Le détail des correspondances des fonctions de commande est donnée dans le listing placé dans le corps de l'article.

PORT[$302]:=$05 déblocage du verrou et préparation de la conversion

$302 est l'adresse où il faut porter le nombre qui code le port C.
$\$05 = 5 = 2^2 + 2^0 = 0101 = 00000101$ codé sur 8 bits. En fait seul les quatres bits PC3 à PC0 sont concernés (en sortie pour l'ordinateur).
PC3 est non connecté, PC2 = 1 = CSL, le verrou d'adresse est «transparent» pendant la phase d'adressage pour permettre d'entrer dans les registres les bits d'adresse ; PC1 = 0 = a0 pour préparer la lecture des bits les plus signifiants par la suite, PC0 = 1 = $R/\overline{C}$, ce qui prépare la conversion qui a lieu lors du passage de $R/\overline{C}$ à 0.

PORT[$300]:= (CE shl 5) + (SD shl 4) + (numero) commande AIS (numéro de voie etc...)

$300 est l'adresse où il faut porter le nombre qui code le port A, d'après la figure 2 il s'agit principalement des commandes concernant le signal analogique.

«numero» est la variable qui contient le numéro de la voie à scruter, soit 12 dans l'exemple.
(numero) est la valeur de « numero» en binaire, soit 1100, mais encore 00001100 sur 8 bits.

SD = 0 dans l'exemple, soit (SD) = 0 ou encore 00000000 sur 8 bits
cela correspond à $\overline{SE}$/D = 0, mode masse commune
(SD shl 4) = 00000000 ce qui correspond à un décalage à gauche de 4 rangs (shift left) qui est invisible dans ce cas !

CE = 1 dans l'exemple, soit (CE) = 00000001
(CE shl 5) = 00100000 après décalage à gauche de 5 rangs.

Le résultat de la somme logique est 00101100 qui est affecté aux fonctions décrites par le port A. On notera que
PA6 = 0 code $\overline{CS}$ valide et que PA7 = 12/$\overline{8}$ = 0 code le mode «lecture en 8 bits» .
PA5 code CE, PA4 code $\overline{SE}$/D et PA3 à PA0 codent le numéro de voie scrutée.

PORT[$302]:=$01 blocage du verrou

Le mot codé sur le port C est 00000001. Cela correspond au passage de CSL à 0 (PC2 = 0) ce qui bloquera le verrou d'adresse pendant le reste du cycle (voie choisie) ; par contre R/$\overline{C}$ reste à 1 (PC0 = 1) pour pouvoir passer à 0 ensuite.

PORT[$302]:=$00 début de conversion

Le mot codé sur le port C est 00000000. Cela correspond au passage de R/$\overline{C}$ et de $\overline{S}$/H à 0 (PC0 = 0), ce qui donne l'ordre de conversion. En conséquence, STS (status) passe à 1 après un certain délai (noté tds) lorsque la conversion effective commence.

UNTIL (PORT[$302] AND $10) = $00 test de fin de conversion

AND $10 revient à faire une multiplication logique par 00010000. Cela revient à considérer que 7 bits sur 8 seront donc déjà systématiquement nuls puisque l'on fait appel à l'opérateur «et» (A ou B renvoie un 0 si A = 0 ou B = 0). Dès lors que PC5 sera nul (PC5 code STS, status),

alors le huitième bit du produit sera aussi nul. C'est ce qui est testé par la valeur $00 = 0 = (00000000).
Le test est donc : «jusqu'à ce que le produit (...) soit nul».

PORT[$302]:=$01 autorisation de lecture et préparation de la prochaine conversion

Le mot codé sur le port C est 00000001, donc R/$\overline{C}$ repasse à 1, c'est donc l'état R indiquant la lecture qui est valide. D'autre part, cette fonction devra être à 1 au cycle suivant pour donner l'ordre de conversion lors de son passage à 0.

MSB:=PORT[$301] lecture des premiers 8 bits les plus signifiants

Il s'agit d'une «entrée» puisqu'une variable est affectée. Ce qui est affecté à la variable MSB est le mot binaire disponible sur le port B (adresse $301).

PORT[$302]:=$03 autorisation de lecture des bits les moins signifiants

$03 = 3 = $2^1 + 2^0$ = (00000011) ce qui code PC1 = 1 donc a_0 passe à 1.
On laisse PC0 = 1 donc R/$\overline{C}$ = 1 lecture autorisée et PC2 = 0 donc CSL = 0 verrou bloqué.

LSB:=PORT[$301] lecture des 8 bits les moins signifiants

Maintenant que a_0 est à 1, les bits manquants sont disponibles sur le port B (quatre sont rebondants), l'ordinateur les lit et les affecte à la variable LSB.

VALEUR:= (MSB SHL 4) + (LSB SHR 4) reconstitution du résultat en 12 bits

msb est PB11 PB10 PB9 PB8 PB7 PB6 PB5 PB4
(msb shl 4) est PB11 PB10 PB9 PB8 PB7 PB6 PB5 PB4 0 0 0 0 décalage à gauche de 4 rangs
lsb est PB3 PB2 PB1 PB0 PB7 PB6 PB5 PB4
(lsb shr 4) est 0 0 0 0 0 0 0 0 PB3 PB2 PB1 PB0 décalage à droite de 4 rangs.

La somme se fait bit à bit et ne donne lieu ici à aucune «retenue».
On obtient bien valeur : = (PB11 PB10 PB9 PB8 PB7 PB6 PB5 PB4 PB3 PB2 PB1 PB0) sur 12 bits.