

Une introduction aux micro contrôleurs

ST6



C.Dupaty
Professeur de génie électrique
Lycée Fourcade 13120 Gardanne
cdupaty@aix-mrs.iufm.fr

Les 6 dernières pages de ce document sont les transparents de cours numérotés de 1 à 6

Fiche contrat (de préparation de séquence)

Brique TECH M2-1 : Technologie et application des produits

Objectifs :

- ⇒ Découvrir les composants micro contrôleurs SGS-THOMSON
- ⇒ Travailler en autonomie
- ⇒ Traduire et utiliser une documentation en anglais
- ⇒ Utiliser un outil de documentation multimédia

Pré requis :

Connaissances et savoir-faire d'un technicien supérieur en électronique ou électrotechnique

A partir de :

TP1 et TP2 de ce document

CDROM : DATA DISK SGS-THOMSON 3^{ième} édition 1996

Starter Kit ST622X

PC Multimédia Equipé des logiciels AST6, LST6, SIMST6 sous WINDOWS 95

Conditions :

Travail individuel, évaluation formative, durée 4h en laboratoire.

Critères d'évaluation :

Toutes les réponses doivent être apportées

Les traductions demandées ne doivent laisser aucune ambiguïté sur le sens du texte, elles doivent être très proche du texte original en anglais.

L'utilisation de l'ordinateur doit être aisée. (manipulation de la souris, des fenêtres ...)

L'écriture sera soignée de manière à pouvoir réutiliser le document réponse

Le travail doit être le plus autonome possible, le professeur étant dans le laboratoire pour guider les stagiaires et résoudre les problèmes matériels

Remarque : *Le lecteur doit compléter les paragraphes précédés du symbole *

Fiche de déroulement de séquence : Durée 4h, 15mn de pose , 9 mn de tolérance

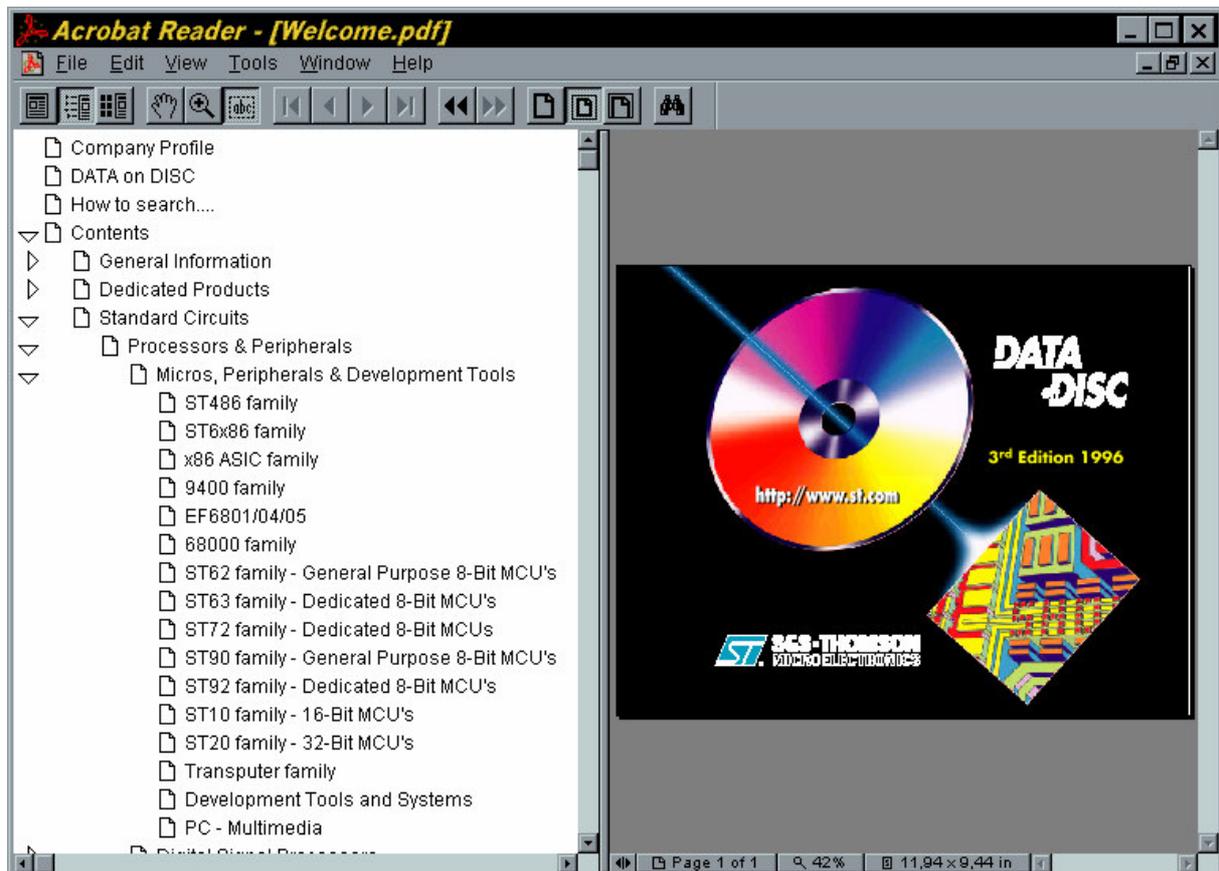
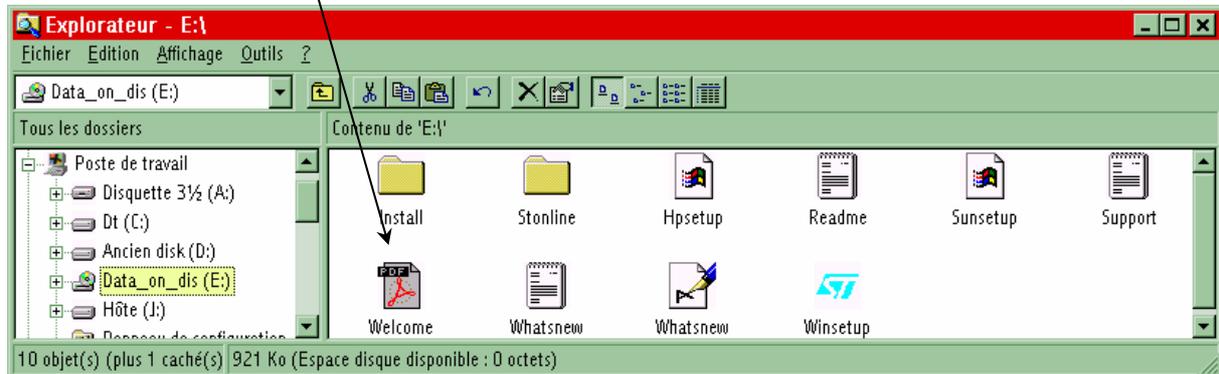
Durées (mn)	Etapes et contenu	Activités du professeur	Activités des stagiaires	Moyens matériels et documents utilisés	Connaissances et savoir-faire apportés
10mn	Présentation du contrat, des conditions de travail	Il présente les objectifs et le contrat, la méthode pédagogique retenue (évaluation formative)	Ils écoutent et peuvent demander des précisions	Fiche contrat Tableau blanc Transparent n°1	
20mn	Contrôle des pré requis	Il questionne oralement et amène les stagiaires à confirmer leurs pré requis sur les microcontrôleurs	Ils mobilisent leurs connaissances afin de répondre aux questions.	Tableau blanc	
1mn		Il distribue les TP et les documents ressources.			
80mn	Apports de savoirs nouveaux et développement des savoir-faire	Il présente le TP n°1 et place les stagiaires en situation d'autonomie. Il les guide, les conseille...	Ils recherchent, participent et notent sur le document réponse. Ils notent les réponses dictées et corrigent éventuellement leur travail	PC, CD ROM TP n°1 Tableau blanc Transparent n°2, 3, 4	Découverte CD ROM SGS, famille ST62xx, unité centrale
60mn	Apports de savoirs nouveaux et développement des savoir-faire	Il présente le TP n°2 et place les stagiaires en situation d'autonomie. Il les guide, les conseille. Ensuite, il inscrit les mots clés au tableau. Il observe le travail des stagiaires, aide ceux en difficulté. Il inscrit les mots clés au tableau, il corrige les erreurs	Ils recherchent, participent et notent sur le document réponse. Ils notent les réponses dictées et corrigent éventuellement leur travail	PC, AST6+SIMST6+ STARTER KIT Transparent 5 Tableau blanc	Prise en main de l'outil STARTER KIT, découverte de l'assembleur AST6, du simulateur SIMST6., des possibilités du micro contrôleur
45mn	Exercice	Il présente le problème, il observe et aide les stagiaires afin de leur permettre de réussir l'exercice	.Ils mobilisent les connaissances nouvelles pour répondre au problème posé, ils s'auto évaluent en référence au contrat de départ	PC, AST6+SIMST6+ STARTER KIT Transparent 6 Tableau blanc	

TP n°1 : Découverte des microcontrôleurs ST62xx

Présentation du TP

A partir du CD-ROM SGS-THMSON « DATA DISK »

- Cliquer deux fois sur WELCOME.PDF puis descendre dans l'arborescence proposée et valider la famille ST62



Ce CDROM permet de consulter tous les produits fabriqués par SGS-THOMSON
La documentation **complète** de la famille ST62 est consultable

Le micro contrôleur ST62xx est disponible en de nombreuses version, ce cours a pour objectif de montrer les potentiels des produits SGS THOMSON à partir d'une analyse superficielle du ST6225B (page 18)

Le lecteur est invité à répondre aux questions posées à partir des moyens décrits ci dessus

1. Caractéristiques des uC ST622x

1.1. Généralités

Qu'est qui différencie le ST6225B du ST62E25B et du ST62T25B ?



ST6225B : Version ROM

ST62E25B : Version EPROM

ST62T25B : Version OTP

Les uC ST sont fabriqués en technologie HCMOS, que signifie ce terme ?



High Speed Complementary Metal Oxide Silicium, c'est la technologie utilisé dans les usines de Rousset

Quel est le principe de fonctionnement d'un transistor MOS ?



La commande s'effectue par une tension qui crée un champ électrique rendant plus ou moins conducteur un barreau de silicium

Quels sont les raisons pour lesquels les fondeurs utilisent plutôt des transistors MOS que bipolaires ?



Les transistors sont plus petits et consomme beaucoup moins d'énergie

Qu'appelle-t-on :

INTERUP VECTOR ?



Vecteur d'interruption, c'est une adresse contenant l'adresse du sous programme d'interruption

ROM (capacité du ST62E25) ?



Read Only Memory, c'est dans cette mémoire qu'est stocké le programme d'application, 4KO pour le ST62E25

RAM (capacité du ST62E25) ?



Random Acces Memory, c'est dans cette mémoire que sont stocké les variables, 64 octets pour le ST62E25

12/20 fully software programmable I/O ?



12 ou 20 broches programmables en entrées ou sortie indépendamment

Pull-up résistor ?



Résistance de « tirage », utilisé pour polariser les sortie à drain ouvert

I/O lines can sink up to 20mA ?



Lignes de sortie pouvant fournir jusqu'à 20 mA

1.2. Constitution

De combien de port parallèles dispose le micro contrôleur ST6225B?

 **20 ports // : PA0-PA7, PB0-PB7, PC4-PC7**

Quels sont les autres périphériques intégrés ?

 **16 ports analogiques avec un C.A.N**

1 timer

1 Chien de garde (Watchdog)

Qu'appelle-t-on 8 BIT CORE ?

 **C'est l'unité centrale (C.P.U) commune à tous les microcontrôleurs de la série ST6**

Expliquer le rôle du bloc « STACK LEVEL » en quoi cette fonction diffère-t-elle des autres micro contrôleurs

 **Le bloc STACK LEVEL est une pile LIFO, qui permet de sauvegarder le contexte lors de l'appel d'un sous programme ou lors du déclenchement d'une interruption. Habituellement la position de la pile système est choisie par le programmeur qui la place en générale en haut de la RAM. Sur le ST6, 6 registres LIFO sont dédiés à cette fonction, il n'est pas nécessaire d'initialiser la pile système**

1.3. Interruptions

Enoncer les différentes sources d'interruption et les adresses de leurs vecteurs



RESET, FFEh, FFFh Initialisation

NMI, FFCh, FFDh Interruption non masquable

PortA, FF6h, FF7h Déclenchement sur front descendant ou sur niveau

PortB, FF4h, FF5h Déclenchement sur front montant ou descendant

PortC, FF4h, FF5h Déclenchement sur front montant ou descendant

(même vecteur PB et PC)

TJMER, FF2h, FF3h Pour TJMER en sortie ou en entrée

CAN, FF0h, FF1h Pour fin de conversion

1.4. Plan mémoire

Le micro contrôleur dispose de deux espaces mémoire, donner leurs adresses, ainsi que le type d'information qui y sont stockées.



PROGRAM SPACE : 0000h à 0FFFh

DATA SPACE : 000h à 0FFh

1.5. Registres :

Quel doit être le contenu du registre IOR pour activer les interruptions sur front descendant sur le port A et front montant sur le port B ?

Poids	D7	D6	D5	D4	D3	D2	D1	D0
Noms		<i>les</i>	<i>esb</i>	<i>gen</i>				
Valeur	x	0	1	1	x	x	x	x

Quel doit être le contenu des registres ORA, DDRA, DRA pour :

- ☞ Placer le port PA0 en entrée avec résistance de pull-up et interruption activée
- ☞ Placer le port PA2 en sortie avec l'option 20mA sink push-pull
- ☞ Placer le port PA4 en entrée analogique

ORA

Poids	D7	D6	D5	D4	D3	D2	D1	D0
Noms				<i>ORA4</i>		<i>ORA2</i>		<i>ORA0</i>
Valeur	x	x	x	1	x	1	x	1

DDRA

Poids	D7	D6	D5	D4	D3	D2	D1	D0
Noms				<i>DDR4</i>		<i>DDR2</i>		<i>DDR0</i>
Valeur	x	x	x	1	x	1	x	1

DRA

Poids	D7	D6	D5	D4	D3	D2	D1	D0
Noms				<i>DRA4</i>		<i>DRA2</i>		<i>DRA0</i>
Valeur	x	x	x	0	x	x	x	1

Expliquer succinctement les quatre fonctions pouvant être remplies par le TIMER

 *Sortie = 0 après un temps t*

Sortie = 1 après un temps t

Mesure de durée d'impulsion à l'état haut

Compteur d'événements

Quel est le rôle du WATCHDOG, comment est-il réarmé ?

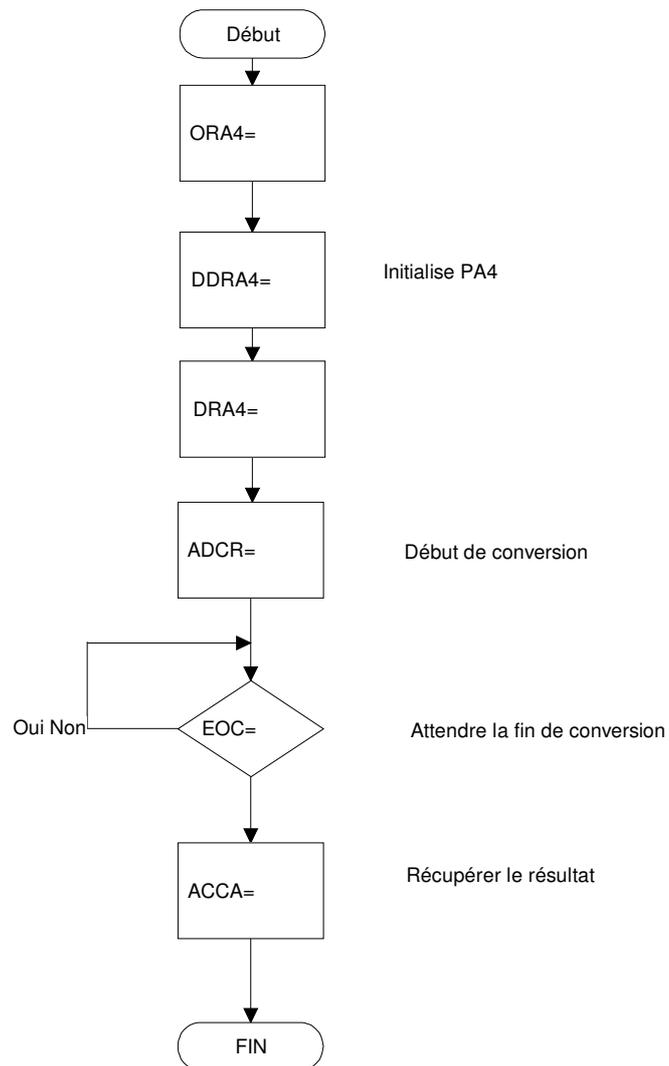


Cette fonction permet de détecter un fonctionnement anormal du programme (plantage), le chien de garde doit être réarmé périodiquement.

Le chien de garde est réarmé en écrivant OFDh à l'adresse 0D8h (DWRD)

Le chien de garde ne peut pas être désactivé après sa mise en fonctionnement

Compléter l'algorithme permettant d'effectuer une mesure sur le port analogique PA4 sans utiliser d'interruption. Le résultat étant placé dans l'accumulateur.



FACULTATIF :

Proposer un organigramme similaire, la fin de conversion déclenchant maintenant une interruption appelant un sous programme



1.6. Logiciel :

Expliquer le rôle des registres X, Y, V, W, A, PC

 X

Registre d'index, contient l'adresse de l'opérande. Adressage Indirect

 Y

Registre d'index, contient l'adresse de l'opérande. Adressage Indirect

 V

Mémoire fixe, contient l'opérande. Adressage Short Direct

 W

Mémoire fixe, contient l'opérande. Adressage Short Direct

 A

Accumulateur, toutes les opérations arithmétiques et logiques s'effectuent entre A et un registre ou une mémoire, le résultat est placé dans A

 PC

Compteur de programme, contient l'adresse de la prochaine instruction

Expliquer succinctement le principe et le domaine d'application des 9 modes d'adressages du micro contrôleur.

 Inhérent

Il n'y a pas d'opérande

 Direct

L'opérande représente l'adresse de la donnée

 Short Direct

L'opérande est dans un des registres X, Y, V, W

 Indirect

L'opérande est l'adresse pointé par X ou Y

 Immediate

L'opérande est la donnée de l'instruction ex : ldi a,01

 Program Counter Relative

Réservé aux instructions de test et de saut. L'opérande représente la distance entre le PC et l'adresse de saut

 Extended

L'opérande représente l'adresse de la donnée en ROM

 Bit Direct

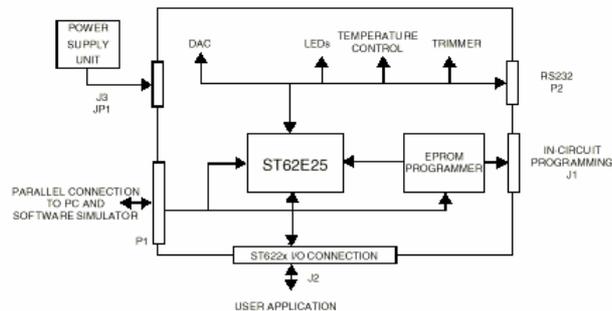
Modification d'un bit d'un donnée

 Bit Direct and Branch

Branchement relatif, conditionnel à l'état d'un bit

2. TP n°2 : Prise en main des outils de développement

➤ A partir de « ST622X-KIT START KIT USER MANUAL »



2.1. Prise en main du KIT

➤ Après lecture des pages 4/45 à 15/45, compléter le tableau ci-dessous :

Broche	Nom	Fonction	Interfaçage
1	VDD	Tension d'alimentation 5v	+5v
2	Timer	E/S Timer	Résistance de chauffage (PWM)
3	OSC1N	Quartz	8MHz
4	OSC0	Quartz	8MHz
5	NMI	Interruption non masquable	
6	PC7	Port //	Interface série asynchrone
7	PC6	Port //	Interface série asynchrone
8	PC5	Port //	Interface série asynchrone
9	PC4	Port //	Interface série asynchrone
10	VPP	Tension prog EPROM	Régulateur VPP78L05
11	RESET	Initialisation	
12	PB7	Port //	Réservé moniteur
13	PB6	Port //	Réservé moniteur
14	PB5	Port //	Réservé moniteur
15	PB4	Port //	Bouton +
16	PB3	Port //	Bouton -
17	PB2	Port //	Réservé moniteur
18	PB1	Port //	Réservé moniteur
19	PB0	Port //	Réservé moniteur
20	PA7	Port //	Réseau R/2R
21	PA6	Port //	Libre
22	PA5	Port //	Potentiomètre
23	PA4	Port //	Thermistance
24	PA3	Port //	LED en entrée
25	PA2	Port //	LED en entrée
26	PA1	Port //	LED en entrée
27	PA0	Port //	LED en entrée
28	VSS	0v	0v

- Connecter le KIT comme indiqué pages 15/45 à 19/45 et le placer dans le mode DEMONSTRATION
- Après lecture des pages 20/45 à 23/45, essayer les différents exemples proposés.
- On remarquera en particulier les possibilités offertes par la fonction C.A.N. La démonstration de l'exemple écrit à l'aide de l'outil FUZZI LOGIC sera étudié en fin de TP.

2.2. Notions d'assembleur

Les pages 35/67 à 39/67 propose un résumé des caractéristiques logicielles du ST62xx

A l'aide de ST6 FAMILY SOFTWARE TOOLS AST6, LST6, SIMST6

- Analyser le programme suivant, traduire oralement la syntaxe de l'assembleur en langage clair et précis.

Les directives d'assemblage sont grisées, les commentaires commencent par un point virgule

ST6 MACRO-ASSEMBLER version 4.40 - May 1995

--- SOURCE FILE : TRAVAIL\FLASH62.ASM ---

```

1      1      ;*****
2      2      ;*
3      3      ;*          ST6 PORTS simple demo  CD 02/1997          *
4      4      ;*          FLASHER sur PORT A                      *
5      5      ;*****
6      6
7      7

8      8      ;|=====|
9      9      ;| Assembly method without linker : ast6 -l -s -m -e example1.asm |
10     10
11     11      ;|=====|
12     12     ;|          Assembly directives          |
13     13
14     14
15     15     .title "ST6 PORTA FLASHER sans IT"
16     16     .vers "st6225"
17     17     .romsize 2
18     18
19     19     ;*****
20     20     ;*          data space          *
21     21     ;|          data registers          |
22     22
23     23      ;|=====|
24     24     .input "travail\6215_reg.asm"
25     25     1 1 ;|-----|
26     26     1 2 ;|          ST6210/15 and ST6220/25 Registers Declaration |
27     27     1 3 ;| Use this file with the .input directive. |
28     28     1 4 ;|-----|
29     29     1 5
30     30     1 6 x .def 80h!m ; Index register.
31     31     1 7 y .def 81h!m ; Index register.
32     32     1 8 v .def 82h ; Short direct register.
33     33     1 9 w .def 83h ; Short direct register.
34     34     1 10
35     35     1 11 a .def 0ffh!m ; Accumulator.
36     36     1 12
37     37     1 13 dra .def 0c0h ; Port a data register.
38     38     1 14 drb .def 0c1h ; Port b data register.
39     39     1 15 drc .def 0c2h ; Port c data register.
40     40     1 16
41     41     1 17 ddra .def 0c4h ; Port a direction register.
42     42     1 18 ddrb .def 0c5h ; Port b direction register.
43     43     1 19 ddrc .def 0c6h ; Port c direction register.
44     44     1 20
45     45     1 21 ior .def 0c8h ; Interrupt option register.
46     46     1 22 drwr .def 0c9h ; Data rom window register.
47     47     1 23
48     48     1 24 ora .def 0cch ; Port a option register.
49     49     1 25 orb .def 0cdh ; Port b option register.
50     50     1 26 orc .def 0ceh ; Port c option register.
51     51     1 27
52     52     1 28 adr .def 0d0h ; A/D data register.
53     53     1 29 adcr .def 0d1h ; A/D control register.
54     54     1 30
55     55     1 31 psc .def 0d2h ; Timer prescaler register.
56     56     1 32 tcr .def 0d3h!m ; Timer counter register.
57     57     1 33 tscr .def 0d4h ; Timer status control register.
58     58     1 34
59     59     1 35 wdr .def 0d8h ; Watchdog register.
60     60     25
61     61     26
62     62     27 ;|=====|
63     63     28 ;| constants definition |
64     64     29

```

```

65          30  exemple  .equ 0abh
66          31
67          32  ;*****
68          33  ;*                PROGRAM SPACE                *
69          34  ;*****
70          35
71          36

;=====|
72          37  ;|                main program                |
73          38

;=====|
74          39
75          40          .org 880h
76 P00 0880 0DC401 P00 0880 41      reset  ldi ddra,01h          ; Port A0 en sortie les autres en entr,e
77 P00 0883 0DCC01 P00 0883 42          ldi ora,01h          ; Standard Push-Pull
78 P00 0886 17AA P00 0886 43          ldi a,0aah          ; par exemple
79 P00 0888 2D P00 0888 44      loop    com a
80 P00 0889 9FC0 P00 0889 45          ld dra,a
81 P00 088B F188 P00 088B 46          call  tempo
82 P00 088D 8988 P00 088D 47          jp loop
83          48  ;
84          49  ;
85 P00 088F 0D8040 P00 088F 50      tempo  ldi  x,040h
86 P00 0892 1D P00 0892 51      encx   dec  x
87 P00 0893 F0 P00 0893 52          jrnz  encx
88 P00 0894 CD P00 0894 53          ret
89          54  ;*****
90          55  ;*                Restart and interrupt Vectors        *
91          56  ;*****
92          57
93          58          .org 0ff0h
94          59
95 P00 0FF0 04 P00 0FF0 60      adc   nop
96 P00 0FF1 4D P00 0FF1 61          reti
97 P00 0FF2 04 P00 0FF2 62      timer nop
98 P00 0FF3 4D P00 0FF3 63          reti
99 P00 0FF4 04 P00 0FF4 64      portb  nop
100 P00 0FF5 4D P00 0FF5 65          reti
101 P00 0FF6 04 P00 0FF6 66      porta  nop
102 P00 0FF7 4D P00 0FF7 67          reti
103 P00 0FF8 04 P00 0FF8 68          nop
104 P00 0FF9 04 P00 0FF9 69          nop
105 P00 0FFA 04 P00 0FFA 70          nop
106 P00 0FFB 04 P00 0FFB 71          nop
107 P00 0FFC 04 P00 0FFC 72      nmi   nop
108 P00 0FFD 4D P00 0FFD 73          reti
    
```

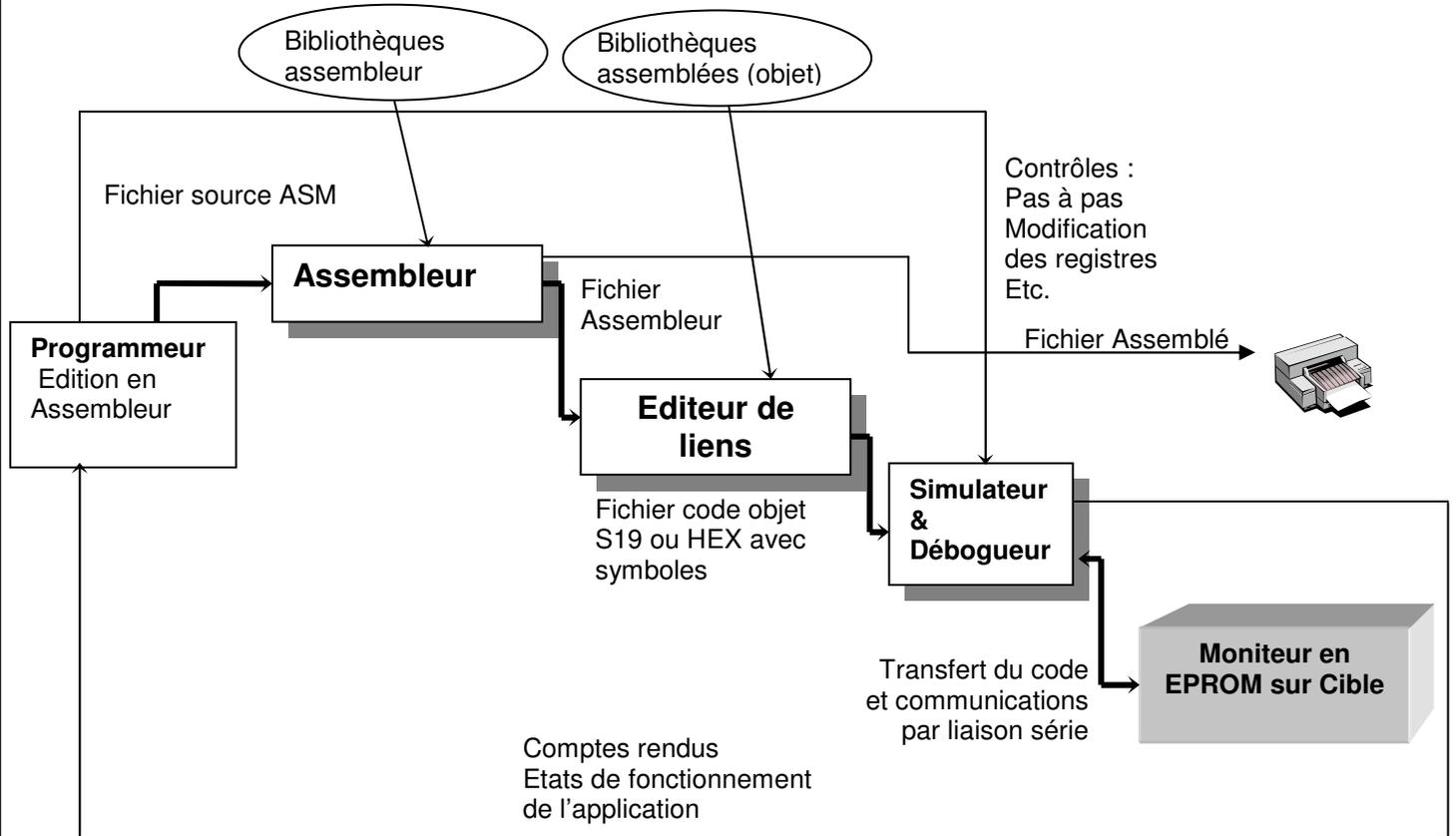
** SPACE 'PAGE_0' SECTION MAP **

name	type	size
PG0_0	TEXT	15
PG0_1	TEXT	E

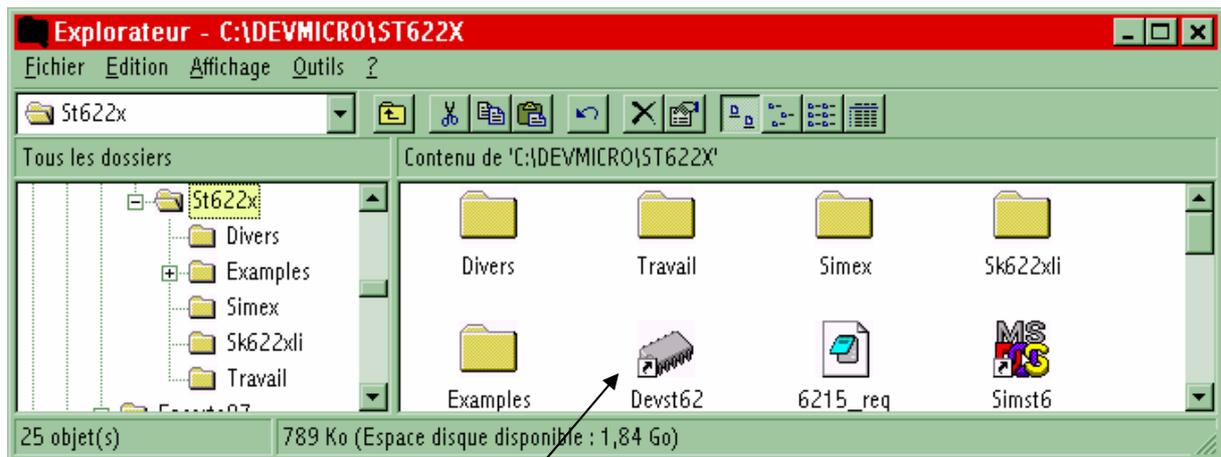
No error detected
No warning

2.3. Le simulateur / Moniteur SIMST6

Structure de l'outil de développement :

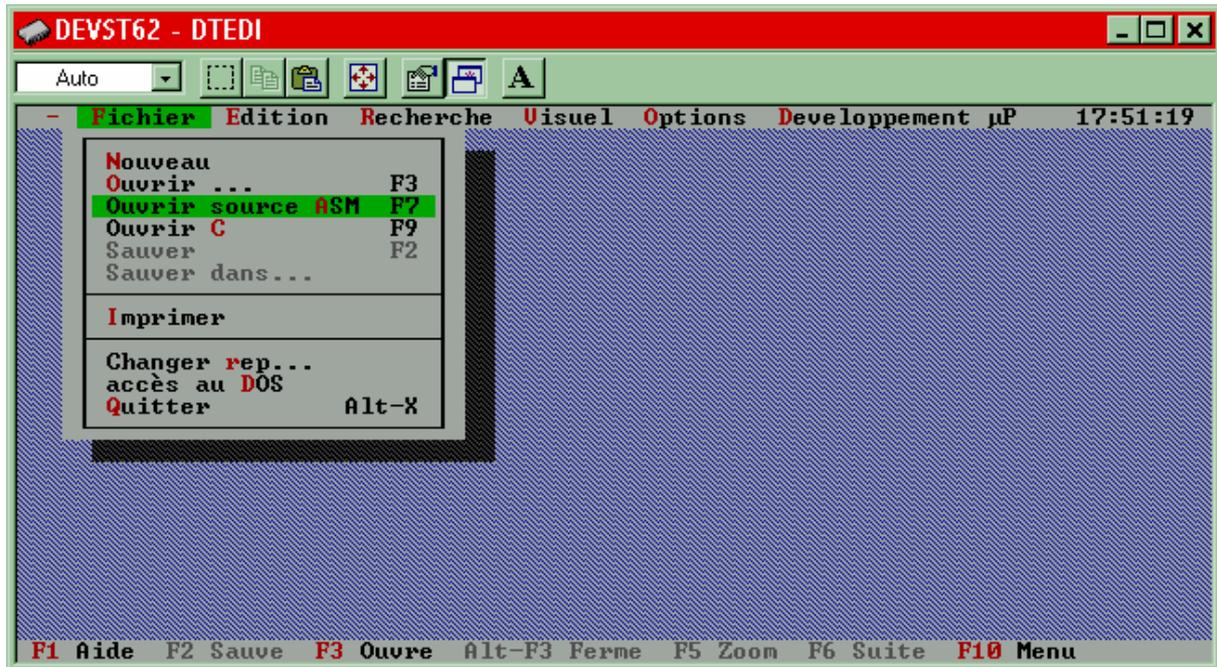


➤ Lancement de l'outil de développement :



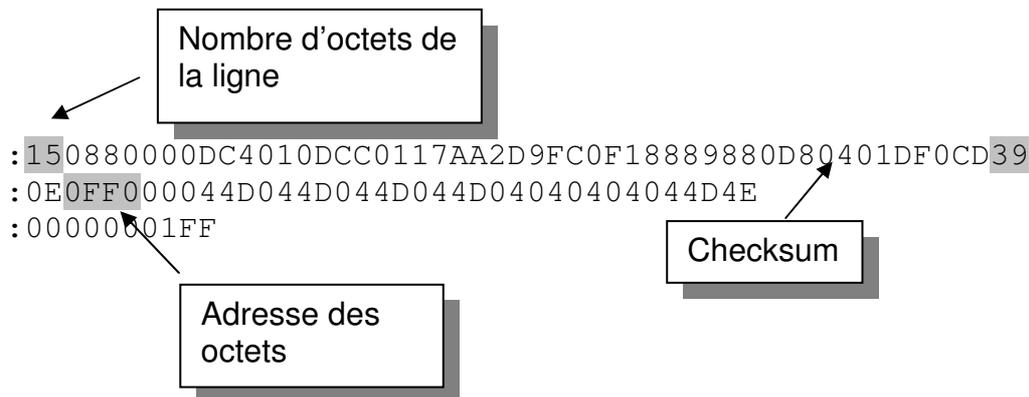
☞ sur cette icône

- L'éditeur de texte est lancé automatiquement, l'écran suivant apparaît, OUVRIER le fichier FLASH62.ASM



Comparer le fichier édité à celui étudié précédemment, puis «développement », puis «assembleur ». Après assemblage, le fichier FLASH62.LIS est automatiquement édité. Vérifier qu'aucune erreur n'est signalée.

Pour information éditer le fichier FLASH62.HEX. par «fichier » «ouvrir »



- « Développement », puis «simulateur », choisir le fichier FLASH62.HEX, le KIT ne doit pas être connecté en mode Simulation. Le fichier HEX est automatiquement chargé en mémoire, on peut le vérifier par «watch » puis «dump PROG » ou «assembleur »

The screenshot shows the DEVST62 - SIMST6 software interface. The 'Registers' window is active, displaying assembly code with addresses and labels:

```

0880 0DC401 reset: ldi dra,01
0883 0DCC01      ldi ora,01
0886 17AA      ldi a,AA
0888 2D        loop: com a
0889 9FC0      ld dra,a
088B F188      call tempo
088D 8988      jp loop
088F 0D8040 tempo: ldi X,40
0892 1D        encx: dec x
0893 F0        jrnz encx
0894 CD        ret
0895 00        jrnz 0896
0896 00        jrnz 0897
0897 00        jrnz 0898
0898 00        jrnz 0899
0899 00        jrnz 089A
  
```

At the bottom of the window, the status bar shows: BA HEX FT OFF BE OFF LOG OFF.

- ⇒ Afin d'effectuer une simulation, positionner le PC en début de programme (adresse : 880H) par « SHIFT F3 » (donner une adresse en RAM), utiliser F1 pour passer d'une fenêtre à l'autre

The screenshot shows the DEVST62 - SIMST6 software interface with the 'DISPLAY' window open. The status bar at the top shows: STACK = 0, NMI = 1, INT = 0, NORM = 0, PC = 0880, Z=0, C=0, Z=0, C=0, Z=0, C=0. The 'Display/Modify' window shows memory addresses and values:

```

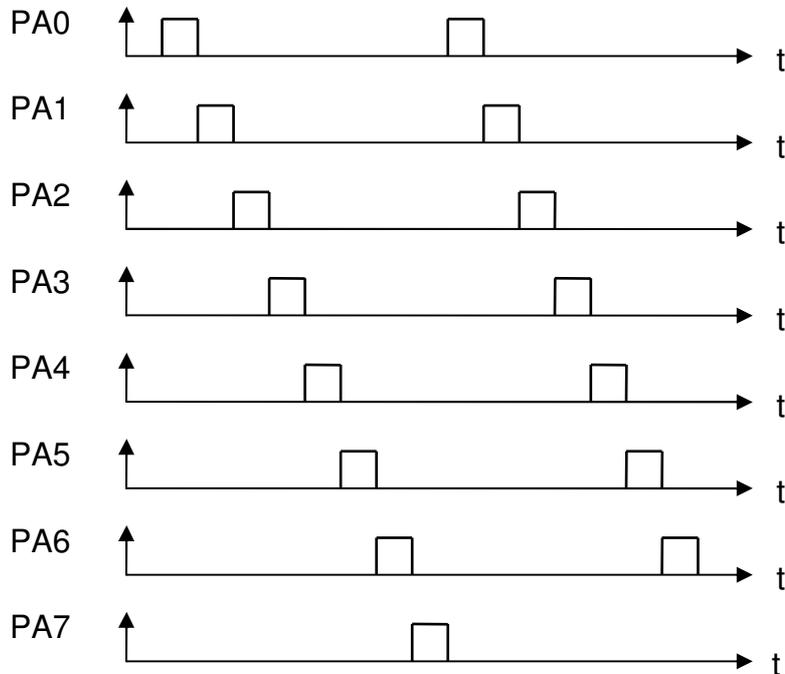
* Display/Modify
from : 0
00A8 00 00 00 00 00 00 00 00 *.....*
00B0 00 00 00 00 00 00 00 00 *.....*
00B8 00 00 00 00 00 00 00 00 *.....*
00C0 00 00 00 00 00 00 00 00 *.....*
00C8 00 00 00 00 00 00 00 00 *.....*
00D0 00 00 7F FF 00 00 00 00 *.....*
  
```

At the bottom of the window, the status bar shows: BK: 0 ADDR: 00A8 NAME: ***** STATUS: ** WR: N. The bottom status bar shows: BA HEX FT OFF BE OFF LOG OFF.

- ⇒ F7 lance l'exécution PAS à PAS, faire dérouler le programme, les longues temporisations peuvent être réduites en modifiant le registre X

2.4. Exercice

- A partir du fichier FLASH62.ASM, écrire le programme CHENILLE.ASM réalisant un «chenillard » sur le port A, la durée de l'état haut n'est pas imposée.
- Effectuer la mise au point à l'aide du simulateur puis les tests à l'aide du KIT (PA0-PA4 uniquement)
- Attention, les fichiers ASM doivent être enregistrer dans le répertoire «travail »



```

*****
* ST6 CHENILLARD ED 11/1997 *
*****

.title "ST6 CHENILLARD SUR PORT A sans JT" ;titre pour fichier LJS
.vers "st6225" ;cible
.romsize 2 ;4KO
.input "travail\6215_reg.asm" ;fichier equivalences RAM

.org 880h ;début ROM
;
reset ldi ddra,OFFh ; Port A en sortie
      ldi ora,OFFh ; Sorties Push-Pull (pour driver LED)
encor rlc a ; efface (C)arry
      ldi a,1h ; Allume LED0
boucle ld dra,a
      rlc a ;rotation
      jrc encor ; si C il faut r,initialiser
      ip boucle
;
.org Offett ; vecteur RESET
ip reset
;

```

Fiche contrat

Brique TECH M2-1 : Technologie et application des produits

Objectifs :

- Découvrir les composants micro contrôleurs SGS-THOMSON
- Travailler en autonomie
- Traduire et utiliser une documentation en anglais
- Utiliser un outil de documentation multimédia

Pré requis :

Connaissances et savoir-faire d'un technicien supérieur en électronique ou électrotechnique

A partir de :

- TP1 et TP2
- CDROM : DATA DISK SGS-THOMSON 3^{ème} édition 1996
- Starter Kit ST622X
- PC Multimédia Equipé des logiciels AST6, LST6, SIMST6 sous WINDOWS 95

Conditions :

Travail individuel, évaluation formative, durée 4h en laboratoire.

Critères d'évaluation :

- Toutes les réponses doivent être apportées
- Les traductions demandées ne doivent laisser aucune ambiguïté sur le sens du texte, elles doivent être très proche du texte original en anglais.
- L'utilisation de l'ordinateur doit être aisée. (manipulation de la souris, des fenêtres ...)
- L'écriture sera soignée de manière à pouvoir réutiliser le document réponse
- Le travail doit être le plus autonome possible, le professeur étant dans le laboratoire pour guider les stagiaires et résoudre les problèmes matériels

Quel doit être le contenu du registre IOR pour activer les interruptions sur front descendant sur le port A et front montant sur le port B ?

Poids	D7	D6	D5	D4	D3	D2	D1	D0
Noms		<i>les</i>	<i>esb</i>	<i>gen</i>				
Valeur	x	0	1	1	x	x	x	x

Quel doit être le contenu des registres ORA, DDRA, DRA pour :

- Placer le port PA0 en entrée avec résistance de pull-up et interruption activée
- Placer le port PA2 en sortie avec l'option 20mA sink push-pull
- Placer le port PA4 en entrée analogique

ORA

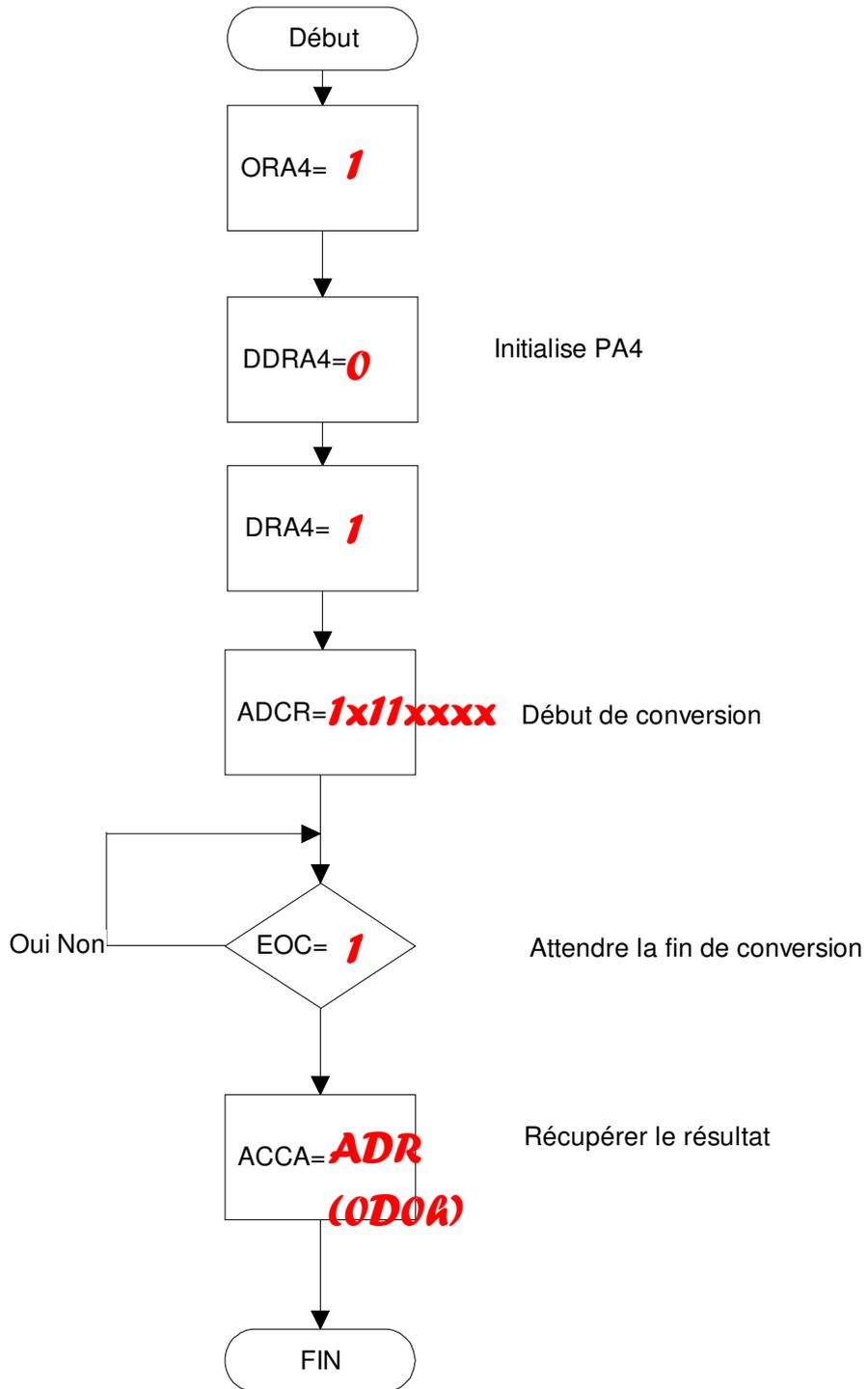
Poids	D7	D6	D5	D4	D3	D2	D1	D0
Noms				<i>ORA4</i>		<i>ORA2</i>		<i>ORA0</i>
Valeur				1		1		1

DDRA

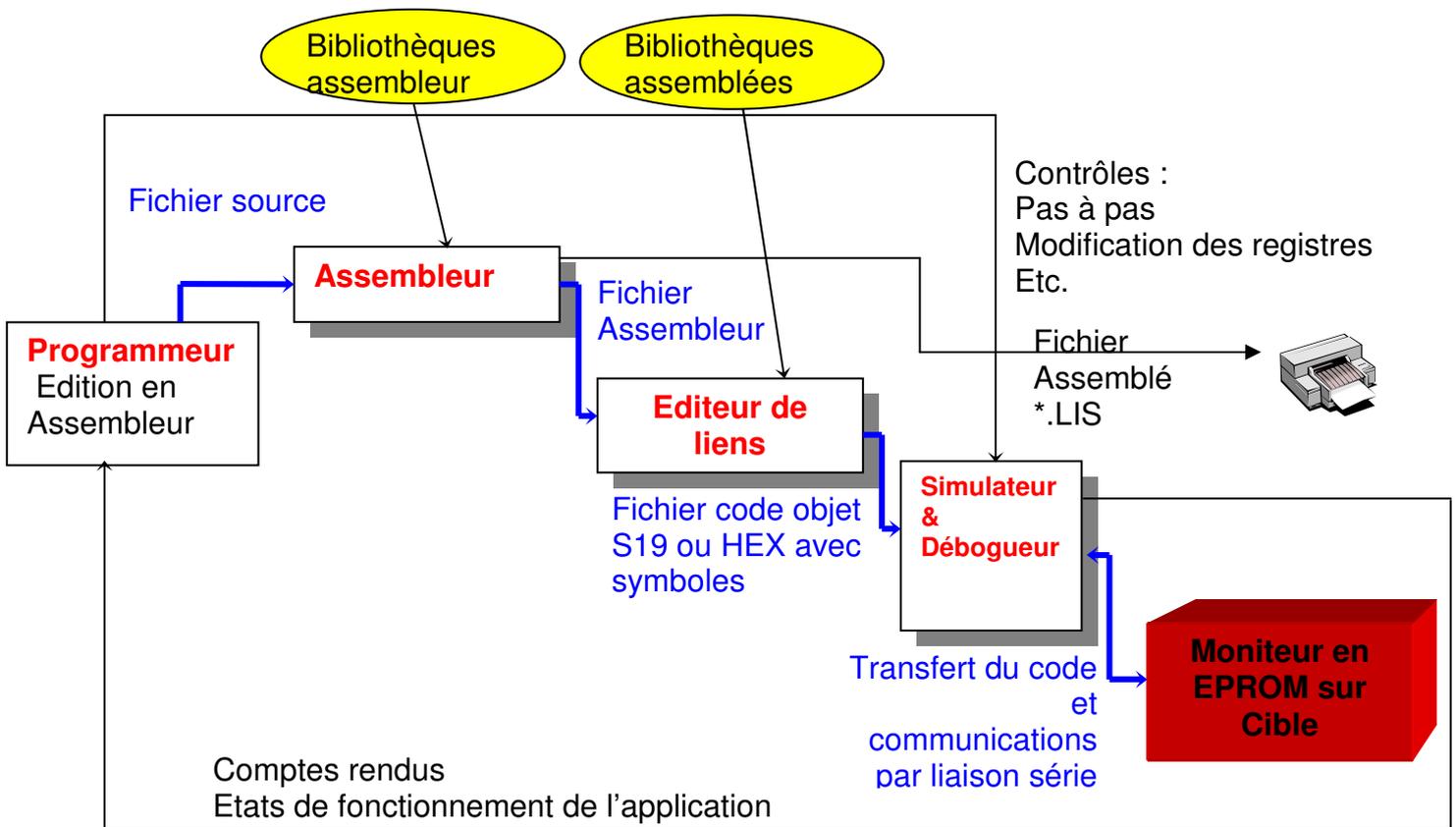
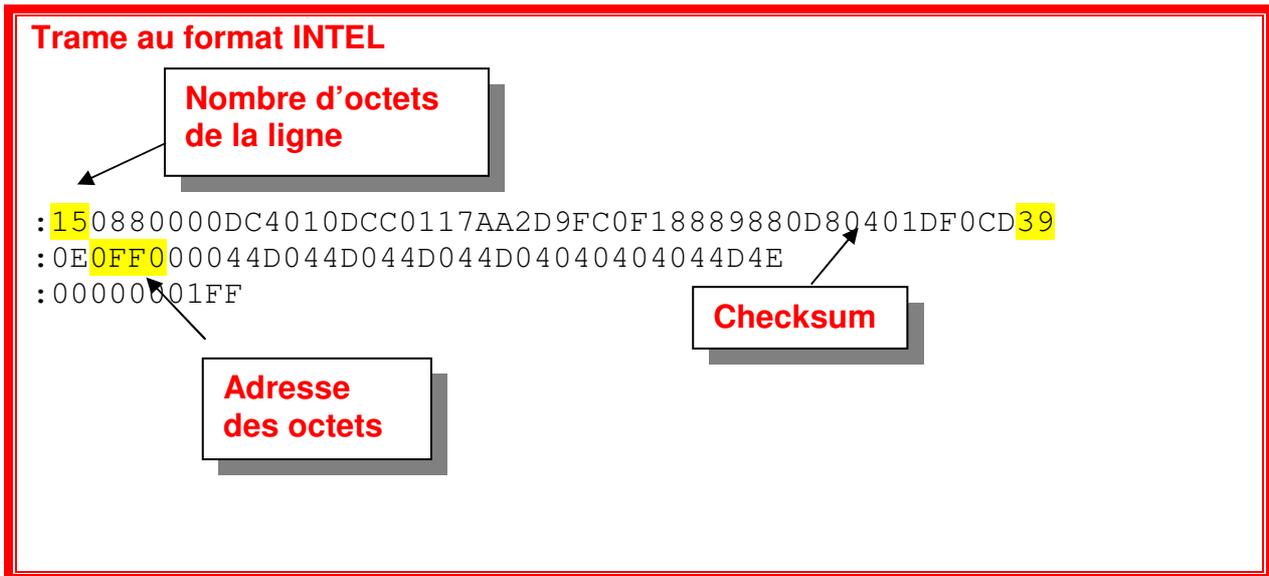
Poids	D7	D6	D5	D4	D3	D2	D1	D0
Noms				<i>DDR4</i>		<i>DDR2</i>		<i>DDR0</i>
Valeur				1		1		1

DRA

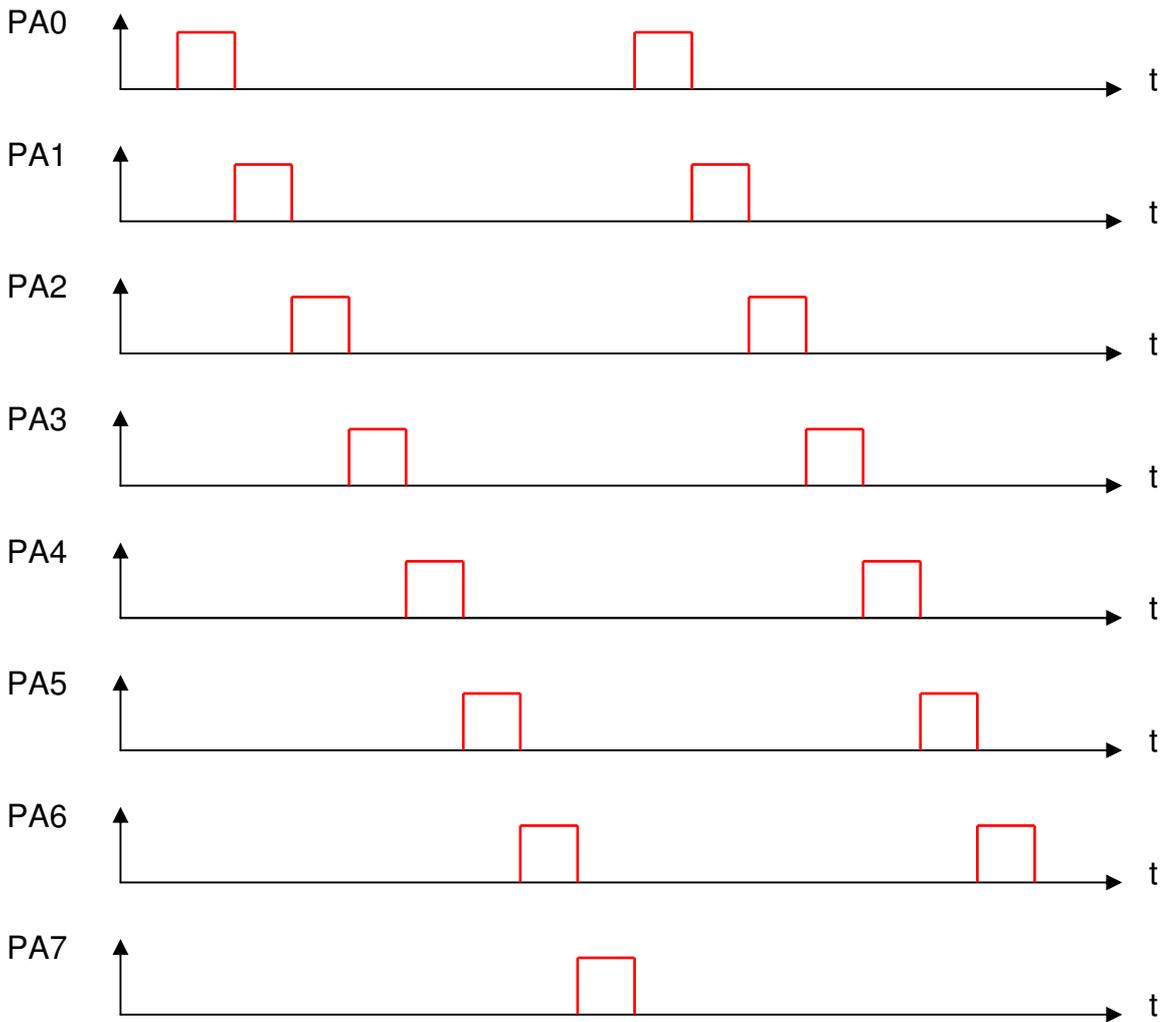
Poids	D7	D6	D5	D4	D3	D2	D1	D0
Noms				<i>DRA4</i>		<i>DRA2</i>		<i>DRA0</i>
Valeur				0		x		1



Broche	Nom	Fonction	Interfaçage
1	VDD	<i>Tension d'alimentation 5v</i>	+5v
2	Timer	<i>E/S Timer</i>	Résistance de chauffage (PWM)
3	OSC1N	<i>Quartz</i>	8MHz
4	OSC0	<i>Quartz</i>	8MHz
5	NMI	<i>Interruption non masquable</i>	
6	PC7	<i>Port //</i>	Interface série asynchrone
7	PC6	<i>Port //</i>	Interface série asynchrone
8	PC5	<i>Port //</i>	Interface série asynchrone
9	PC4	<i>Port //</i>	Interface série asynchrone
10	VPP	Tension prog EPROM	Régulateur VPP78L05
11	RESET	<i>Initialisation</i>	
12	PB7	<i>Port //</i>	Réservé moniteur
13	PB6	<i>Port //</i>	Réservé moniteur
14	PB5	<i>Port //</i>	Réservé moniteur
15	PB4	<i>Port //</i>	Bouton +
16	PB3	<i>Port //</i>	Bouton -
17	PB2	<i>Port //</i>	Réservé moniteur
18	PB1	<i>Port //</i>	Réservé moniteur
19	PB0	<i>Port //</i>	Réservé moniteur
20	PA7	<i>Port //</i>	Réseau R/2R
21	PA6	<i>Port //</i>	Libre
22	PA5	<i>Port //</i>	Potentiomètre
23	PA4	<i>Port //</i>	Thermistance
24	PA3	<i>Port //</i>	LED en entrée
25	PA2	<i>Port //</i>	LED en entrée
26	PA1	<i>Port //</i>	LED en entrée
27	PA0	Port //	LED en entrée
28	VSS	<i>0v</i>	0v



La durée de l'état haut n'est pas imposée.



```

*****
;
;*          ST6 CHENILLARD   CD 11/1997          *
*****
;
        .title      "ST6 CHENILLARD SUR PORT A sans IT" ;titre
        .vers       "st6225"                       ;cible
        .romsize    2                               ;4KO
        .input      "travail\6215_reg.asm"          ;fichier equivalences
RAM      .org        880h                           ;début ROM
reset   .ldi        ddra,0FFh                       ; Port A en sortie
        .ldi        ora,0FFh                       ; Sorties Push-Pull
encor   .rlc        a                               ; efface (C)arry
        .ldi        a,1h                            ; Allume LED0
boucle  .ld         dra,a
        .rlc        a                               ;rotation
        .jrc        encor                          ; si C il faut réinitialiser
        .jp         boucle

;
        .org        0ffeH                           ; vecteur RESET
        .jp         reset

```