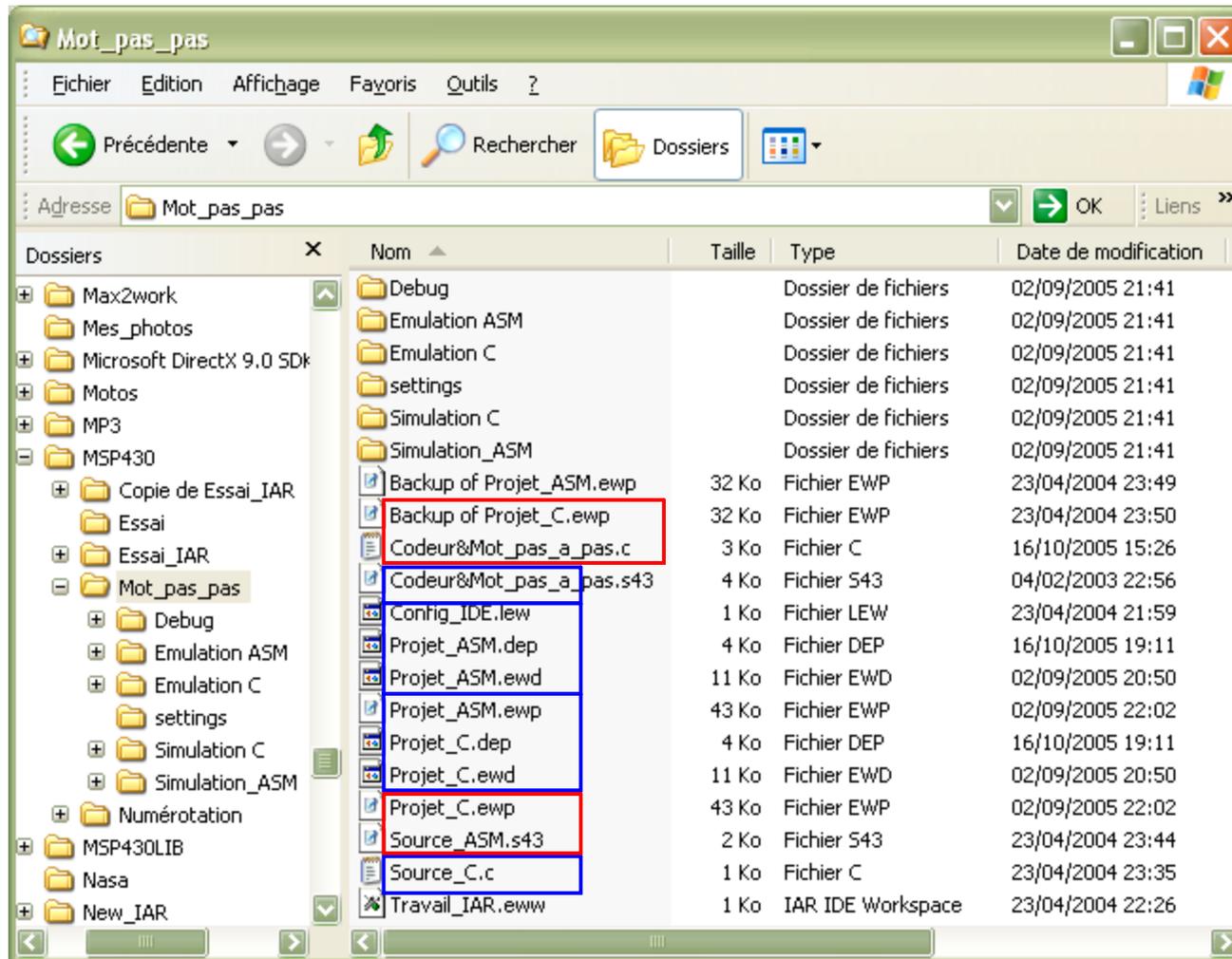


AIDE MÉMOIRE ENVIRONNEMENT IAR V4

1. Démarrer un nouveau projet

- Ouvrir le répertoire D:\MSP430
- Dupliquer le dossier Projet IAR **et le renommer** (choisir un nom représentatif de l'application)
- Contenu typique du nouveau dossier (nommé ici Mot_pas_pas)
- Renommer le fichier source : Source_C.c et/ou Source_ASM.s43 suivant l'application



Fichiers importants à sauvegarder :

- Travail_IAR.eww : "workspace" ou plan de travail
- Projet_ASM.* : Options de l'environnement de développement en mode "assembleur"
- Projet_C.* : Options de l'environnement de développement en mode "C"
- Config_IDE.lew : Configuration de l'environnement de développement (fenêtres)
- *.s43 : sources écrits en "assembleur"
- *.c : Sources écrits en "C"

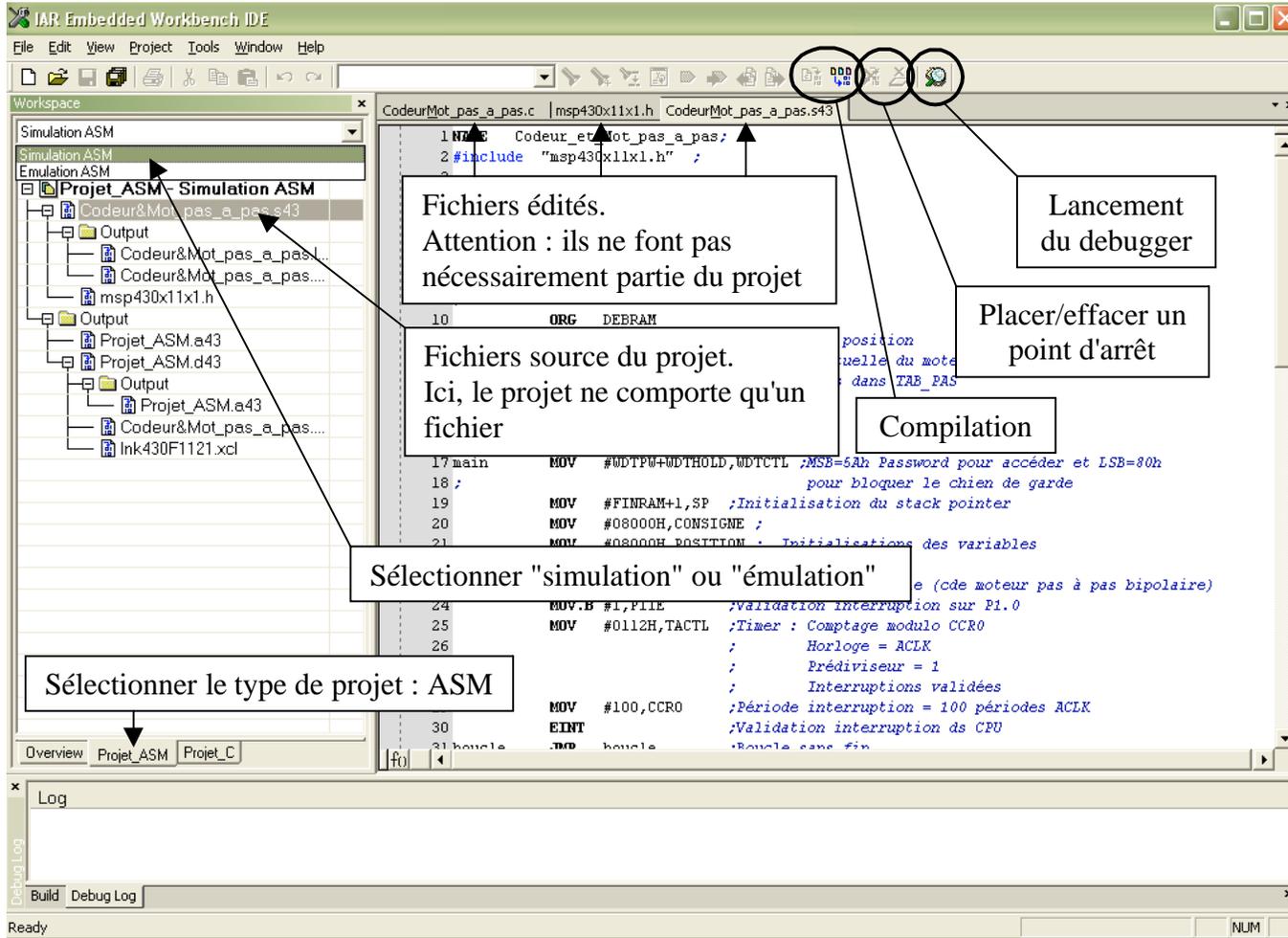
Démarrer :

- Lancer l'environnement IAR V4 en cliquant sur le raccourci "IAR V4 "
- Ouvrir le plan de travail (workspace) : D:\MSP430\\Travail_IAR.eww
- Clic droit sur le nom du projet pour ajouter les fichiers sources du projet et éliminer "Source_C.c"
- Double-cliquer sur le fichier "source" (voir page suivante). Il s'ouvre à droite
- Sauvegarder l'ensemble du dossier en fin de session pour retrouver la configuration à la prochaine session.

2. Ouvrir un ancien projet

- Ouvrir le dossier D : \MSP430
- Y copier le dossier sauve lors de la dernière session
- Lancer l'environnement IAR V4 en cliquant sur le raccourci "IAR V4 "
- Ouvrir le plan de travail (workspace) : D:\MSP430\

3. Environnement typique pour un projet écrit en "assembleur"



L'éditeur de fichier ASM formate le texte de la façon suivante :

- instructions et directives reconnues : **en gras**
- commentaires et directives # : **en bleu**

L'assembleur accepte les instructions en majuscules et minuscules

Le projet peut être compilé de 2 façons :

- Compile : seul l'assembleur est lancé. Le fichier objet (.d43) n'est pas créé.
- Build : l'assembleur et l'éditeur de liens sont lancés. Le fichier .d43 est créé.

Le lancement du debugger est toujours précédé d'une compilation de type "Build".

4. Écran du "debugger" typique pour un projet écrit en "assembleur"

Quitter et fermer le debugger

Contrôles du programme :
 reset ,
 pas à pas : F10 ou ,
 pas à pas "sous-programme" : F11 ou ,
 exécution jusqu'au curseur : ,
 run , jusqu'au point d'arrêt
 Stop 

Indique la prochaine instruction exécutée

**Observation des registres CPU, Timer, Ports, ...
 Choix par les onglets**

Register

P0	= 0
P1	= 0
P2	= 0
P3	= 0
P4	= 0
P5	= 0
P6	= 0
P7	= 0
P1DIR	= 0x00
P1IFG	= 0x00
P1IES	= 0x00
P1IE	= 0x01
P1SEL	= 0x00
P2IN	= 0x00
P2OUT	= 0x00
P2DIR	= 0x0F
P2IFG	= 0x00
P2IES	= 0x00
P2IE	= 0x00
P2SEL	= 0x00

Memory

0200	00 80 00 80 00 63 cc bf bb e5 e5 0d 7f 20 8d 80C..... ..
0210	3b bb 1e 29 14 0b a5 1f 3e b7 95 7d e7 14 a8 3d	...>...>...>...=
0220	db 39 da 62 ee 60 1e 6c e0 3f 28 79 d2 43 06 23	..9..b...l.?(y.C.#
0230	fe f2 d0 68 c4 56 ad 40 44 38 fd 22 98 0f 57 9a	..h.V.@D8."..W.
0240	e1 da bc 41 b8 6e 28 a4 43 3f d7 f6 e7 48 f3 a1	..A.n(.C?...H..
0250	fa ab 06 7e cd 05 4c 61 f7 1f e6 ad d9 09 e7 9c	...~.La.....
0260	91 b6 4b 0a 76 21 55 cc 58 a4 59 79 02 7f 7f 26	..K.v!U.X.Yy.!!&
0270	48 b6 ef f8 a7 49 06 19 b9 eb ed e4 7b 3a db 61	H...I.....{..a
0280	2f 1e b2 59 e8 4e 40 2a 6a 32 03 98 79 80 02 44	/.Y.N@*j2...y..D
0290	d5 e6 b4 86 60 1d 11 60 3c a3 26 b2 5c 1b e7 ebX.&.\...

En mode "émulation", le MSP430 doit évidemment être connecté au PC via la liaison d'émulation JTAG.

5. Environnement typique pour un projet écrit en "C"

The screenshot displays the IAR Embedded Workbench IDE interface. The main window shows a C source file named 'CodeurMot_pas_a_pas.c' with the following code:

```

1 //*****
2 // MSP-FET430F1121- Codeur incrémental et moteur pas à pas *
3 //*****
4
5 #include "msp430.h"
6
7 #define TAB_PAS1 = (8, 2, 4, 1);
8 int IND_PAS;
9 const char TAB_PAS1 = (8, 2, 4, 1);
10
11 // Fonction : Incrémenter ou décrémenteur CONSIGNE
12 // Paramètres :
13 // - CONSIGNE : Valeur à incrémenter ou décrémenteur
14 // - PORT1_VECTOR : Adresse du port 1 (ici P1.0)
15 #pragma vector=PORT1_VECTOR
16 __interrupt__ void PORT1_ISR(void)
17 {
18     PI1ES ^= 1; // Inverser le flanc actif de P1.0 = A
19     if (PI1IN & 1 == 1)
20     { // Flanc montant sur A
21         if (PI1IN & 2 == 2) CONSIGNE = CONSIGNE + 1; // Test du sens de rotation
22         else
23             CONSIGNE = CONSIGNE - 1;
24     }
25     else
26     { // Flanc montant sur A
27         if (PI1IN & 2 == 2) CONSIGNE = CONSIGNE - 1; // Test du sens de rotation
28     }
29 }

```

Annotations and their locations:

- Fichier édité. Attention : il ne fait pas nécessairement partie du projet**: Points to the file 'CodeurMot_pas_a_pas.c' in the workspace tree.
- Fichiers source du projet. Ici, le projet ne comporte qu'un fichier**: Points to the 'CodeurMot_pas_a_pas.c' file in the workspace tree.
- Sélectionner "simulation" ou "émulation"**: Points to the 'Simulation C' dropdown menu in the workspace.
- Sélectionner le type de projet**: Points to the 'Proj_C' button in the bottom toolbar.
- Compilation**: Points to the 'Build' icon in the toolbar.
- Placer/effacer un point d'arrêt**: Points to the 'Breakpoint' icon in the toolbar.
- Lancement du debugger**: Points to the 'Run' icon in the toolbar.

The Messages window at the bottom shows the linker output:

```

IAR Universal Linker V4.590/386
Copyright 1987-2005 IAR Systems. All rights reserved.

232 bytes of CODE memory
86 bytes of DATA memory (+ 12 absolute)
24 bytes of CONST memory

```

6. Écran du "debugger" typique pour un projet écrit en "C"

Placer/effacer un point d'arrêt sur le curseur

Quitter et fermer le debugger

Contrôles du programme :
 reset ,
 pas à pas : F10 ou ,
 pas à pas "sous-programme" : F11 ou ,
 exécution jusqu'au curseur : ,
 run , jusqu'au point d'arrêt
 Stop 

Observation des registres CPU, Timer, Ports, ...
 Choix par les onglets

Indique la prochaine instruction exécutée

Observation des variables (glisser / coller)

Ports	Expression	Value
P1IN = 0x00	CONSIGNE	-32768
P1OUT = 0x00	POSITION	-32768
P0 = 0	IND_PAS	0
P1 = 0	CCR0	30000
P2 = 0		
P3 = 0		
P4 = 0		
P5 = 0		
P6 = 0		
P7 = 0		
P1DIR = 0x00		
P1IFG = 0x00		
P1IES = 0x00		
P1IE = 0x01		
P1SEL = 0x00		
P2IN = 0x00		
P2OUT = 0x00		
P2DIR = 0x0F		
P2IFG = 0x00		
P2IES = 0x00		
P2IE = 0x00		
P2SEL = 0x00		

Points d'arrêt :
 en mode "run",
 l'exécution du
 programme s'arrête
 quand le programme
 atteint le point d'arrêt.