

# Résumé des Activités en séance TP 1

## Premiers pas sur MicroVision

## Premiers programmes en Assembleur 8051

### RAPPEL : Objectifs globaux des séances pratiques

- **Séance 1 : Premiers pas sur MicroVision – Premiers programmes simples en assembleur 8051**

- Séance 2 : Codage de sous-programmes et utilisation de la pile
- Séances 3, 4 et 5 :
- Séances 6 :

#### 1. Objectifs de la séance

- Apprentissage à l'utilisation de l'environnement de développement Microvision – Codage – Simulation – Débogage
- Codage en assembleur – Utilisation du jeu d'instructions – Connaissance de l'architecture mémoire du 8051

#### 2. Sources documentaires

- Chapitre du cours consacré au 8051
- Poly : « Documentation Architecture 8051 et Microcontrôleur 8051F020 »
- Pour aller plus loin : Documentations E-campus : « Reference 8051 Instruction Set » et « Datasheet 8051F020 »
- Vidéos sur la chaîne YouTube CPE-ELN3

#### 3. Préparation de la séance

- Lecture des chapitres « Présentation de l'architecture 8051 », « Cœur 8051 » et « jeu d'instruction 8051 » du document « Documentation Architecture 8051 et Microcontrôleur 8051F020 » (voir sur le E-campus)
- A l'aide de toutes les documentations mises à disposition, codage sur papier des exercices 1, 2 et 5 du fichier Exo\_SM1

#### 4. Rendus de préparation

- Pas de rendu demandé.

#### 5. Déroulement de la séance

Les activités sont classées par ordre chronologique.

- DECOUVRIR l'environnement de développement MicroVision.
- TESTER les différentes fonctionnalités de MicroVision à l'aide d'un code exemple.
- CODER en assembleur, au travers d'exercices simples
- COMPRENDRE les différents espaces mémoires du 8051
- ASSEMBLER et SIMULER et DEBOGUER les programmes assembleur réalisés.

#### 6. Rendus de fin de séance

Ces rendus seront faits sur le e-campus.

- Le fichier d'exercices Exo\_SM1.asm.

#### 7. Compétences acquises

A l'issue de ce TP, vous devez pouvoir répondre par l'affirmative à ces 6 questions.

- Je sais créer un projet sous Microvision pour mettre en œuvre le 8051F020, et je sais démarrer une session de débogage en simulation.
- Je sais assembler un programme, le charger sur la carte, et l'exécuter en mode pas à pas ou avec des points d'arrêts.
- Je sais utiliser le débogueur pour visualiser le contenu des mémoires et des registres et je sais éventuellement modifier leur contenu.
- Je sais identifier et utiliser les différents modes d'adressage lorsque je code en assembleur (registre, direct, indirect, immédiat, bit, indexé, relatif...).
- Je sais utiliser tout le jeu d'instruction du 8051, en consultant la documentation mise à votre disposition.
- Je connais les différents espaces mémoires du 8051 et connais les instructions et les modes d'adressage y accéder.

Si ne pouvez pas répondre par l'affirmative à toutes ces questions, les objectifs du TP ne sont pas atteints, vous devez donc veiller à combler ces lacunes avant la prochaine séance de TP.

## Déroulement de la séance TP 1

### 1. Déroulement de la séance.

#### 1.1. Prise en main de l'IDE Microvision– Utilisation d'un programme source existant.

Pour cette partie, se reporter au Poly : « Documentation Architecture 8051 et Microcontrôleur 8051 » et plus particulièrement au chapitre : « Initiation à Microvision », qui vous permettra de prendre en main cet outil au travers de la mise en œuvre d'un code exemple.

Pour découvrir Microvision, nous vous conseillons d'utiliser un code assembleur de démonstration : **Demo\_TP\_SM1\_V2.asm**. Une vidéo dont le lien sera donné sur le E-campus (chaîne YouTube CPE-ELN3) vous permettra de vous aider à appréhender les fonctionnalités de base de simulation/Débogage.

#### Activités :

- Création d'un projet.
- Assemblage et édition de liens d'un programme d'essai.
- Exécution du programme en simulation.
- Débogage. Exécution en pas à pas ou avec points d'arrêt.
- Visualisation des registres et de la mémoire.

#### 1.2. Exercices en assembleur.

Après être sorti du mode Debug, insérer le fichier **Exo\_SM1.asm** dans le projet, tout en enlevant le fichier **Demo\_tp\_SM1\_V2.asm**. Faites les exercices proposés dans ce fichier. Ils sont nombreux, mais pour la plupart très brefs. Ils permettent de se familiariser avec le langage assembleur du 8051.

Complétez le fichier en ajoutant vos codes assembleur. Utilisez le débogueur pour exécuter le code en pas à pas et pour visualiser les registres et les différentes zone mémoire

#### Notions abordées :

- Les directives d'assemblage de base, propres au programme assembleur A51.
- Les registres de base : ACC, B, DPTR, PSW, PC, R0...R7.
- les instructions de transfert de données et les modes d'adressage associés.
- Les instructions logiques et de manipulation de Bits.
- Les instructions arithmétiques.
- Les instructions de sauts conditionnels et inconditionnels.

**Tous les fichiers à utiliser peuvent être téléchargés depuis le E-campus.**

**Contenu du fichier Exo\_SM1.asm**
**Remarque :**

Pour chacun des exercices suivants, **il faut impérativement vérifier le bon fonctionnement**. Assemblez le code, testez le à l'aide du débogueur, et corrigez-le si besoin. Vérifier les différents registres (PC, DPTR, Ri, PSW, etc.) et les différentes zones mémoires (IDATA, XDATA, CODE).

**Précision 1 :** en assembleur 8051, on exprime un nombre en base 16, en ajoutant le suffixe H et un nombre en base 2, en ajoutant le suffixe B.

**Précision 2 :** en assembleur, toute valeur numérique doit commencer par un chiffre, aussi on écrit 0FFh au lieu de FFH

**EXO 1 : INSTRUCTIONS BASIQUES DE TRANSFERT DE DONNEES**

1. Initialiser l'accumulateur à 0 (coder les 3 solutions possibles), le registre R0 à FFh et la case d'adresse mémoire 40h de la mémoire RAM interne à 55h.
2. Copier le contenu de l'accumulateur dans le registre R1.
3. Copier le contenu de la mémoire 40h de la RAM interne dans la case 42h de la RAM Interne. Coder de 3 manières différentes en utilisant les divers modes d'adressage
4. Copier le contenu de la mémoire 40h de la RAM interne dans la case D0h de la RAM Interne à accès indirect. Que se passe-t-il si on utilise par erreur l'adressage direct sur D0h?

**EXO 2 : MANIPULATION DES ESPACES MEMOIRES**

1. Copier le contenu de 20h (DATA) dans la case mémoire 2FFh en mémoire RAM externe (XDATA).
2. Copier le contenu de 0000h (CODE) dans la case mémoire 82h en mémoire RAM interne.
3. Copier le contenu de la mémoire 0100h de la XDATA dans la case 43h de la RAM Interne.
4. Copier le contenu de la mémoire 1234h de la mémoire CODE dans la case 0102h de la mémoire XDATA.

**EXO 3 : INSTRUCTIONS ARITHMETIQUES - Arithmétique non signée**

1. Incrémenter l'accumulateur de 1.
2. Décrémenter de 1 l'octet d'adresse 33h de la RAM Interne.
3. Echanger le contenu de B avec le contenu de la mémoire 07FFh en mémoire externe (XDATA).
4. En une ligne de code, échanger le contenu de A avec le contenu de la mémoire 60h.

**EXO 4 : INSTRUCTIONS LOGIQUES - MANIPULATION de BITS**

1. Complémenter le **bit** d'adresse 10h. Ou se trouve-t-il ? (c'est le bit X de l'octet d'adresse YY dans la mémoire DATA).
2. Mettre à 1, les bits 0 et 7 de l'octet d'adresse 22h de la RAM interne (sans changer les autres bits).
3. Mettre à 1, les bits 0 et 7 du registre R0 (sans changer les autres bits).
4. Mettre à 1, les bits 0 et 7 de l'octet d'adresse 07ffh de la XDATA (sans changer les autres bits)

**EXO 5 : INSTRUCTIONS DE SAUT CONDITIONNEL ET INCONDITIONNEL**

1. Placez une valeur quelconque dans R3, incrémentez la jusqu'à ce qu'elle atteigne la valeur B6h (pensez à initialiser R3).
2. Remplir la mémoire RAM interne de l'adresse 20h à 40h avec des codes ASCII égrainant l'alphabet.
3. Lire le contenu de l'adresse 0000h à 0002h dans l'espace code et le copier à l'adresse 0000h à 0002h de l'espace XDATA.

**EXO 6 : MANIPULATION DES BANCS de REGISTRES R0-R7**

1. Commuter le banc de registres R0-R7 sur le banc 2 et copier le contenu de A dans le registre R4 sans utiliser "R4" dans l'instruction.
2. Mettre à zéro la case d'adresse 08h de la ram interne en utilisant un adressage par registre (utiliser un registre R0...R7).

**EXO 7 : INSTRUCTIONS LOGIQUES - MANUIPULATION de BITS - 2**

1. Décaler le registre B d'une case vers la gauche, le bit6 devient bit7, le bit5 devient bit6, etc. le bit7 devient bit0.
2. Exécuter une rotation logique gauche sur le DPTR, le bit 15 devient le bit 0. (Initialiser auparavant le DPTR avec la valeur 0F0Fh).

**EXO 8 : INSTRUCTIONS ARITHMETIQUES - Arithmétique non signée - 2**

1. Incrémenter de 2 la case mémoire RAM externe d'adresse 100h.
2. Multiplier les données en RAM interne d'adresse 22h et 23h. Placer le résultat en 24h(LSB) et 25h(MSB).

**EXO 9 : INSTRUCTIONS DE SAUT CONDITIONNEL ET INCONDITIONNEL - 2**

1. Remplir la mémoire RAM externe (XDATA) de l'adresse 120h à 140h avec des codes ASCII égrainant l'alphabet.
2. Incrémentez le DPTR initialisé avec une valeur quelconque jusqu'à la valeur de 0ABCDh.

**EXO 10 : Exercices d'approfondissement - FACULTATIF**

1. Additionner les registres R6 et R7, stocker le résultat dans R5. Quelles sont les limitations ? Pour s'affranchir des limitations précédentes, refaire la même opération mais stocker le résultat dans R4 (LSB) et R5(MSB).
2. Faire l'opération R0 moins R1 et placer le résultat dans R7. Que se passe-t'il si  $R1 > R0$  ?
3. Complémenter le demi-octet de poids faible de l'adresse 2Ah de la RAM interne.
4. Placez une valeur quelconque dans R3, décrémente-la, jusqu'à ce qu'elle atteigne une valeur contenue dans R6 (pensez à initialiser R3 et R6).
5. Initialiser les registres R4 et R5 respectivement avec les valeurs 0Fh et F5h. Faites un OU Exclusif entre ces deux registres et placez le résultat dans l'accumulateur.
6. Stocker le demi-octet de poids faible de la case mémoire d'adresse 44h de la RAM interne dans les deux demi-octets du registre R4 (si la case 44h contient 17h, alors R4 devra contenir 77h).
7. Elever au carré le contenu de l'octet 12h, placer le résultat dans R0 (pds fort) et R1 (pds faible).
8. Soit 2 nombres BCD de 16 bits (valeur 0 à 9999) stockés respectivement dans R0(MSB)-R1(LSB) et R2(MSB)-R3(LSB). Faire l'addition BCD de ces deux nombres, et stocker le résultat dans R4(MSB)-R5(LSB). La retenue sera dans le bit Carry du registre PSW.
9. Compter le nombre de bits à 1 dans le registre B et stocker le résultat dans R5.
10. Diviser le contenu du registre R7 par le nombre 10, placer le quotient dans R5 et le reste dans R6.