

PHY4501 — RIDE

1. ACCÈS DONNÉES

1.1. ÉNONCÉ

Afin d'étudier l'accès aux différents espaces de stockage du microcontrôleur (mémoire de données et SFR, voir figure 1), on souhaite transférer :

- la valeur hexadécimale A4 successivement :
 - dans l'accumulateur ;
 - sur le port P0 ;
 - dans le registre R0 de la banque 0 ;
 - dans le registre R0 de la banque 1 ;
 - dans la variable Adro_1 (adresse mémoire octet 10h) ;
- le contenu de la case mémoire pointée par le registre R1 de la banque 1 dans la variable Adro_2 (adresse mémoire octet 31h) ;
- la valeur 1 dans la variable Adb (adresse mémoire bit 31h).

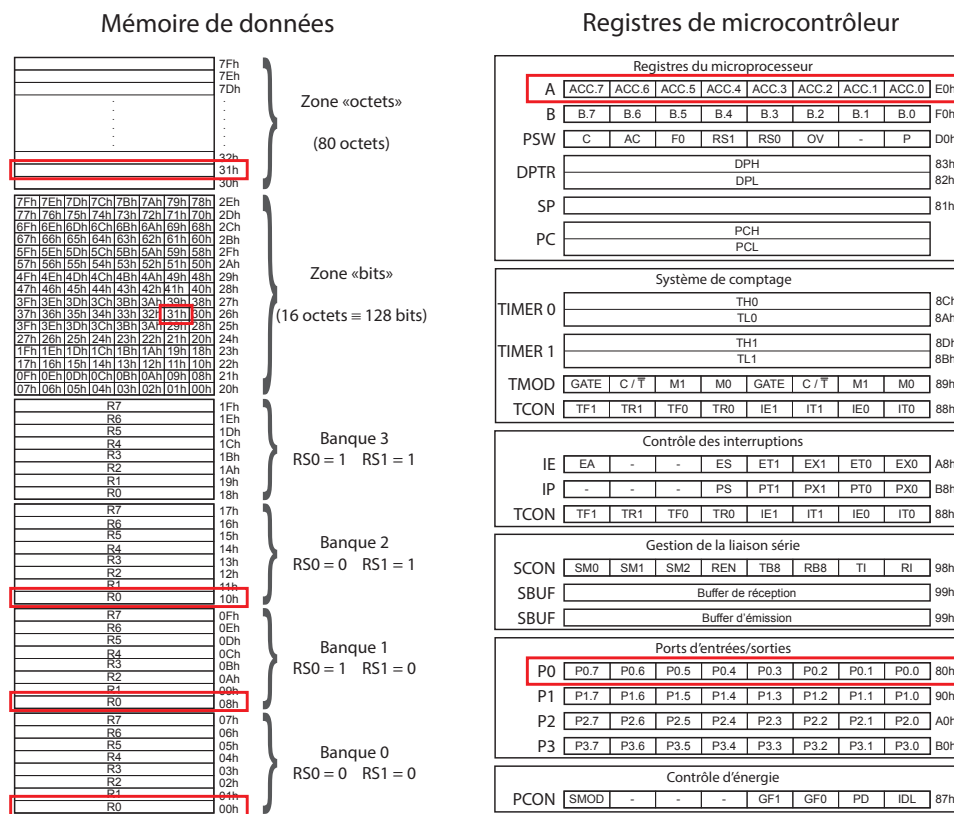


FIGURE 1 – Différents accès de la mémoire du microcontrôleur

1.2. ÉTUDE THÉORIQUE

Q1. Faire le tableau de description des données.

Q2. Écrire, en pseudo-codes, l'algorithme du programme *Donnees*.

1.3. PRÉPARATION DE RIDE

1. Exécuter le logiciel Ride.
2. Créer un projet *TP_mC* et l'enregistrer dans le répertoire *C:\TP\PHY4501\TP*.
3. Créer un fichier *Donnees* et l'enregistrer dans le même répertoire.
4. Associer le fichier au projet.
5. Configurer le projet afin que les noms des SFR soient connus par l'assembleur.
6. Configurer le débogueur.

1.4. SIMULATION

Q3. Sous Ride, écrire le programme, de l'algorithme précédemment défini, en langage assembleur AT89C51, en respectant les consignes suivantes :

- utiliser des directives d'assemblage afin de manipuler des symboles à la place de valeurs numériques ;
- le programme ne doit pas se situer dans la table des vecteurs d'IT ;
- reporter les pseudos-codes en commentaires.

Passer en mode débogueur et ouvrir les fenêtres utiles : *Main Registers*, *Data* et *Bit*. Puis, exécuter le programme pas à pas afin de répondre aux questions ci-dessous.

Q4. Expliquer la valeur du PC pour les quatre premières instructions.

Q5. À chaque instruction, observer les changements au niveau des registres SFR (fenêtre *Main registers*) et de la mémoire de données.

Q6. Détailler le transfert de la donnée dans la variable *Adro_2*.

Q7. À l'aide de la fenêtre *Code view*, commenter le code opératoire et la partie auxiliaire de la dernière instruction du programme lorsque celle-ci est *SJMP fin* puis *LJMP fin*. Donner la zone de mémoire programme accessible par ces deux mnémoniques (*SJMP* et *LJMP*).

2. COMMANDE D'UNE DEL

2.1. ÉNONCÉ

On souhaite commander l'allumage et l'extinction d'une DEL (*Aff*) à chaque front descendant d'un bouton poussoir (*BP*), selon le câblage donné en figure 2.

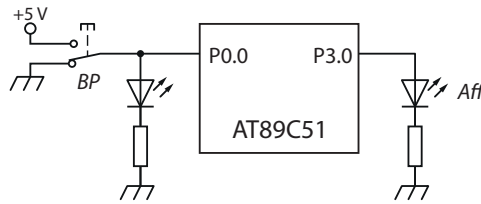


FIGURE 2 – Schéma de câblage de la commande d'une DEL

2.2. ÉTUDE THÉORIQUE

Q1. Faire le tableau de description des données.

L'AT89C51 ne permet pas la détection de fronts sur ces ports d'entrées/sorties. Un front doit donc être traité comme la suite de deux niveaux logiques successifs qu'il faut attendre consécutivement. L'attente peut être vue comme une structure itérative (*Répéter jusqu'à*, par exemple), sans instruction en son sein et où la condition correspond au test du niveau *ad hoc*.

Q2. Écrire, en pseudo-codes, l'algorithme du programme BP_LED.

2.3. PRÉPARATION DE RIDE

1. Dans le projet TP_mC, supprimer l'association du fichier *Donnees*.
2. Créer un fichier BP_LED et l'enregistrer dans le répertoire du projet.
3. Associer le fichier au projet.

2.4. SIMULATION

Q3. Sous Ride, écrire le programme en langage assembleur AT89C51.

En mode débogage, placer un point d'arrêt sur la dernière instruction et ouvrir les fenêtres *Port 1* et *Port 3*.

Q4. Exécuter le programme en mode continu afin de vérifier son bon fonctionnement.

2.5. EXPÉRIMENTATION

1. Ouvrir le lecteur réseau \\fuji\electronique.
2. Ouvrir le répertoire correspondant à la salle (A304 ou A305).
3. Ouvrir le répertoire PosteX ou X correspond à votre numéro de poste.
4. Copier dans le répertoire ouvert, le fichier TP_mC.hex de votre répertoire de travail.
5. Ouvrir le support à insertion nulle et récupérer le microcontrôleur (repérer son sens dans le support).
6. Programmer le microcontrôleur : demander à un enseignant.

Q5. Expliquer le rôle du *checksum*.

Câbler le montage expérimental selon la figure 2 et tester votre programme.

Q6. Dresser un tableau qui donne la correspondance entre l'état logique de sortie (0 ou 1), la tension de sortie et l'état de *Aff* (éteint ou allumé).

3. COMPTEUR

3.1. ÉNONCÉ

On souhaite réaliser un compteur sur quatre bits, selon le schéma de câblage donné en figure 3. Le changement d'état du compteur sera provoqué par un front descendant de *BP*. L'interrupteur *SEL* permet de sélectionner le mode décomptage (*SEL* = 0) ou comptage (*SEL* = 1). L'affichage (*Aff*) sera présenté sous la forme de quatre DEL et d'un afficheur sept segments.

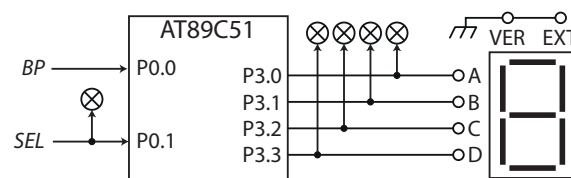


FIGURE 3 – Schéma de câblage du compteur

3.2. TRAVAIL À RÉALISER

- Q1.** Faire le tableau de description des données.
- Q2.** Écrire, en pseudo-codes, l'algorithme du programme *CTR_4b*.
- Q3.** Écrire le programme en langage assembleur AT89C51.
- Q4.** En mode débogage, vérifier le bon fonctionnement du programme.

Programmer le microcontrôleur et effectuer le câblage selon la figure 3.

- Q5.** À l'aide des afficheurs, donner la table de correspondance des bases 2, 10 et 16 pour les 16 premiers symboles.