

BTS INFORMATIQUE ET RESEAUX POUR L'INDUSTRIE ET LES SERVICES TECHNIQUES

Session 2006

EPREUVE E.4 Etude d'un système informatisé

COMMANDE AUTOMATISEE DES CENTRALES HYDRAULIQUES DU RHIN

Corrigé

B. Etude

B1 . Calcul de marnage & puissance

Question B1.1 : Plages horaires

Heure de début	Heure de fin	Nature de l'écluse
00H00	07H00	Rétention
07H00	09H00	équilibre
09H00	14H00	Lâcher
14H00	17H30	équilibre
17H30	20H30	Lâcher
20H30	23H00	équilibre
23H00	00H00	Rétention

Question B1.2 : Marnage

$$hauteur = \frac{Volume}{Longueur \cdot largeur} = \frac{Débit \cdot Temps}{Longueur \cdot largeur} = \frac{37 \cdot 7 \cdot 3600}{14300 \cdot 135} = 0,48 \text{ m}$$

Question B.1.3 Puissance électrique

$$Pe = Pm \cdot \eta_T \cdot \eta_G = Q \cdot g \cdot h \cdot 1000 \cdot \eta_T \cdot \eta_G = 240 \cdot 9,81 \cdot 14,26 \cdot 1000 \cdot 0,85 \cdot 0,94 = 26,83 \text{ Mw}$$

B2. Répartition des niveaux à Kembs :

Question B.2.1 Le PHV donne au PA une consigne de 970 m³/s.

La combinaison est une combinaison **maximale** car c'est un début d'éclusee.

Les débits pour les 6 groupes sont : 245 240 240 245 0 0

Question B.2.2 Le PHV donne au PA une consigne de 980 m³/s.

La combinaison est une combinaison **courante**

Les débits pour les 6 groupes sont : 245 245 245 245 0 0

Question B.2.3 Le PHV donne au PA une consigne de 990 m³/s.

La combinaison est une combinaison **adjacente**

Les débits pour les 6 groupes sont : 85 240 240 185 240 0

Question B.2.4 L'opérateur du PA attribue une consigne de 100 m³/s au groupe 4.

La page chargée est = $(990 - 100) / 10 = 89$. Dans l'algorithme, on ignore les combinaisons utilisant le groupe 4.

La combinaison ne peut donc être une combinaison **courante**, c'est une combinaison **adjacente**

Les débits pour les 6 groupes sont : 170 240 240 100 240 0

La réponse : 170 240 240 0 240 0 est également acceptée.

B3. Liaison fibre optique :

Question B.3.1 Caractéristiques de la fibre optique

Avantages:	Inconvénients :
Immunité aux interférences électromagnétiques Sécurité (pas de repiquage). Débit élevé Isolation galvanique	Coût élevé, mise en œuvre ardue.

B4. Synchronisation horaire :

Question B.4.1 modes RS-232, RS-485 & RS-422

Liaisons	Topologie		Mode		Distance	
	Point à point	Multipoint	Différentiel	Unipolaire	< 100 m	> 1000 m
RS-232	x			x	x	
RS-422	X (1)	X (2)	x			x
RS-485		x	x			x

Remarques concernant la RS422 :

Les 2 réponses sont acceptées car :

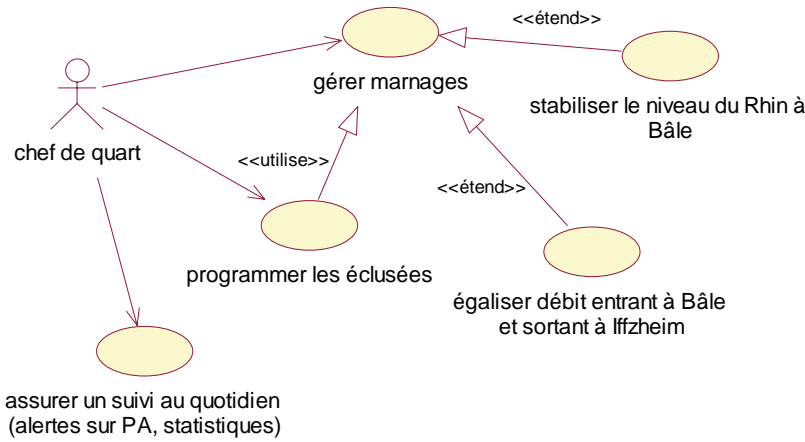
- (1) Une liaison RS422 est essentiellement utilisée pour allonger une liaison RS232 et est considérée à ce titre comme une liaison Point à Point.
- (2) Une liaison RS422 est considérée comme une fausse liaison multipoint (simplex) : dans ce cas un seul équipement est connecté au bus et peut transmettre à tous les autres.

Question B.4.2 Temps de transmission

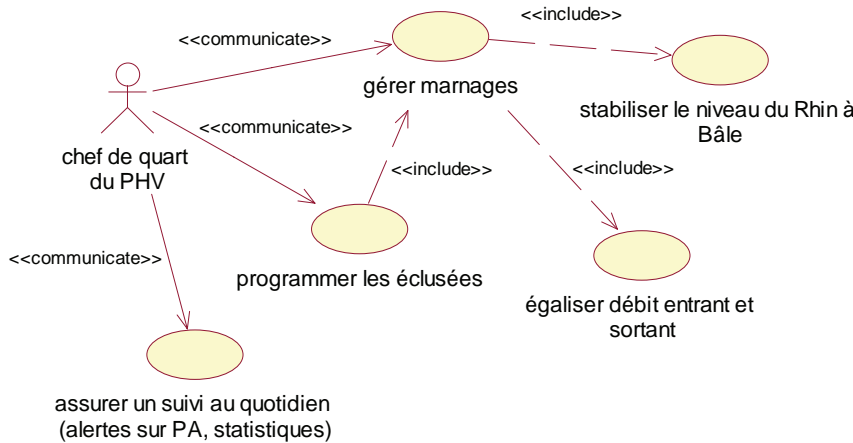
Tps = $(13(\text{Format J}) + 2(\text{CR} + \text{LF})) * (1 + 7 + 1 + 1) / 9600 = 15,5 \text{ ms}$

Question C.2.1 Diagramme des Cas d'Utilisation

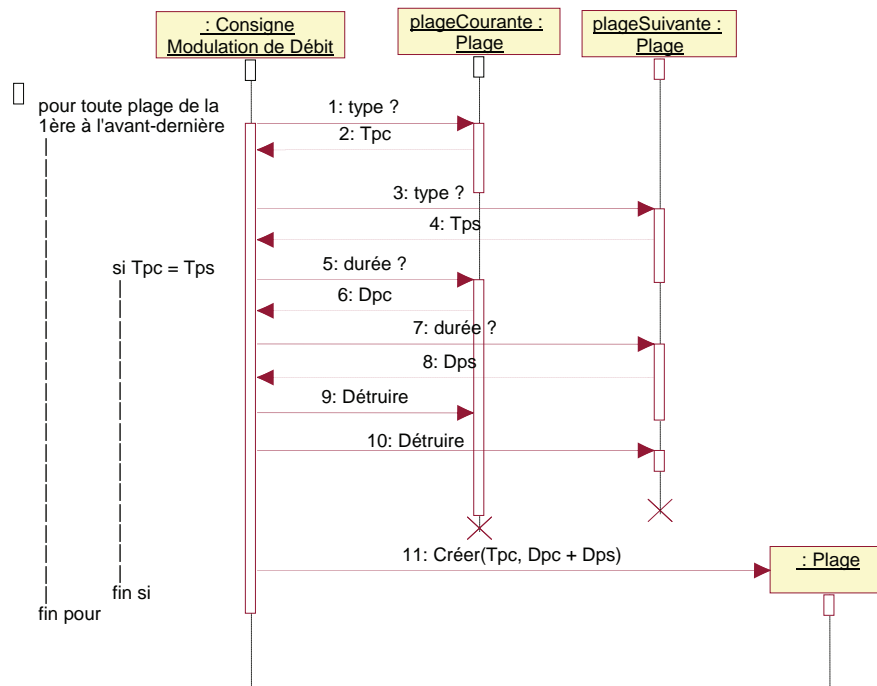
réponse possible en UML 1.1



réponse en UML 2



Question C.3.1 Diagramme de Séquence (scénario de regroupement des plages)



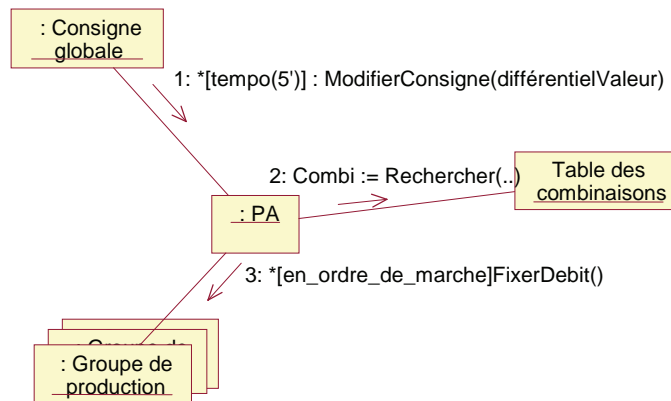
Question C.4.1 Nature synchrone/asynchrone des messages (scénario de régulation du débit) :

- Messages No. 2, 4 et 6 : Appels de méthode de l'objet lui-même (réflexif) donc synchrone ;
 - Messages No. 1 et 3 : Question/réponse donc handshake bloquant, donc synchrone ;
 - Message No. 7 : Itération ;
 - Seul le message No. 5 est asynchrone car échangé entre deux processus évidemment distincts puisqu'ils sont exécutés sur deux calculateurs différents PHV d'une part et PA de l'autre.
- Caractéristiques : Périodique (5 minutes), suivi d'un acquittement à réception.

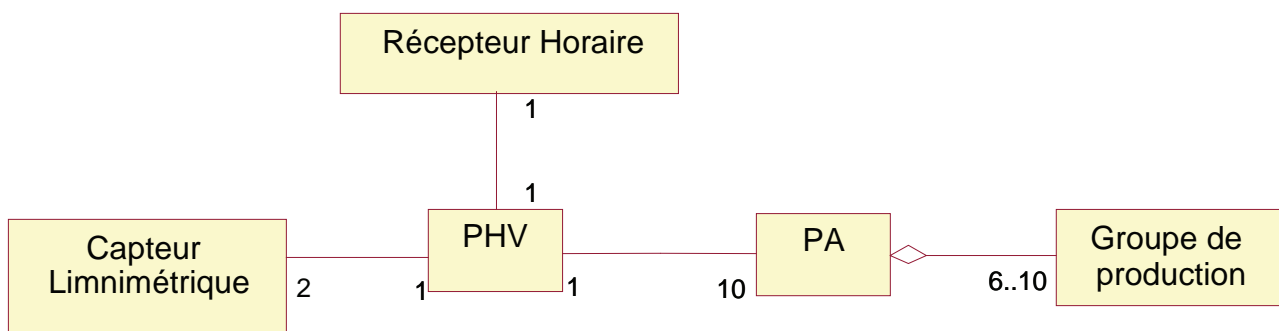
Question C.4.2 Signification du message *[en_ordre_de_marche]FixerDebit()

Pour tous les groupes de production en ordre de marche, appeler la fonction FixerDébit()

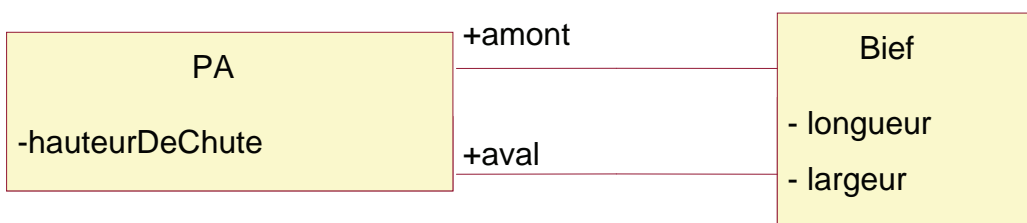
Question C.4.3 Numérotation des messages échangés par les objets



Question C.5.1 Indication des cardinalités sur Diagramme de Classes



Question C.5.2 Diagramme de Classes issu du Diagramme d'Objets



Question C.5.3

Nature de la relation entre les classes Message, Consigne_débit_total et Informations_remontées :

Généralisation ou héritage. Les classes Consigne_débit_total et Informations_remontées sont deux classes spécialisées d'une classe Message plus générale (super-classe)

Cela signifie que ...

un message est spécialisable soit en une consigne de débit total, soit en une somme d'informations remontées ;

une consigne de débit total correspond aux informations remontées;

un message est composé à la fois d'une consigne de débit total et des informations remontées correspondantes.

Question C.5.4

Une Consigne_débit_total est donc composée de ...

D'une Valeur_Débit_Naturel et d'une Valeur_Modulation_Débit. Le lien de composition est représenté par un losange attaché à la classe composite ou agrégée.

Question C.5.5 La contrainte *{ordonnée}* ne serait plus nécessaire si ...

Chaque plage de la collection pouvait être qualifiée par son heure de début et /ou son heure de fin. Cela demande au sein de la classe Plage à ce que soit ajouté un attribut heure à l'attribut durée déjà existant. On peut aussi imaginer que celui-ci soit simplement remplacé par deux nouveaux attributs représentant respectivement l'heure de début et celle de fin de la plage.

D1 .Conception :

Question D.1.1 Déclaration de la classe CKaplan

```
class CKaplan : public CGroupe
{
public:
    CKaplan(int noGroupe);
    ~CKaplan();
    void fixeDebit(int debit);
private:
    void positionnePales (int debit);
};
```

Question D.1.2 La classe CGroupe est abstraite

Car elle contient une méthode virtuelle pure
 virtual void fixeDebit (int)=0;

Les conséquences sont :

- On ne peut instancier d'objets de la classe **CGroupe**.
- La méthode **fixeDebit** est obligatoire au niveau des classes dérivées.

Question D.1.3 Rôle du pointeur this

Le pointeur **this** est le pointeur de l'objet courant ; il permet ici que l'attribut et le paramètre puisse avoir le même nom.

Question D.1.4 Tableau des méthodes

Méthode appelée \ Méthode(s) exécutée(s)	classe CGroupe	classe CKaplan
Constructeur	x	x
Méthode fixeDebit		x
Méthode getResultats	x	

Question D.1.5 Lien entre la classe CPa et la classe CCombinaison

```
class CPa
{
private:
    CCombinaison combinaison;
    int nbGroupes;
    CGroupe **groupes;
    .....
public:
    CPa( CPhv *, int, int *, int);
    ~CPa(void);
    .....
};
```

Question D.1.6 Nature de l'attribut typeTurbine de la classe CCroupe

private : accès interdit depuis une classe dérivée.

protected : accès autorisé depuis une classe dérivée.

typeTurbine doit être protected pour être accessible depuis les classes dérivées.

D2 .Codage :

Question D.2.1 Réception de la consigne

```

while (1)
{
tMessagePa * message = (tMessagePa *)messaging.retirer();
//*****
// Réception message consignes du PHV
//*****
if (message->origineModif == PHV)
{
int debitConsigne = message->messageDuPhv.debitNaturel
+ message->messageDuPhv.debitModulation;

// les paramètres de la méthode sont la consigne globale et le booléen indiquant un début d'écluse.
combinaison.fixeDebit (debitConsigne, message->messageDuPhv.debutOuFinPhase);

Ligne c = combinaison.demandeCombinaison();
repartitDebit(c);
}
//*****
// Réception message de l'opérateur du PA
//*****
.....
} // fin while

```

Question D.2.2 Taille de la structure **28 octets**

Question D.2.3 Instruction C++ donnant la taille : **sizeof(tMessagePa)** ou **sizeof(*message)**

Question D.2.4 Pseudo code des fonctions envoie & retire : **Version 1 :**

Envoie	Retire
P(SemaVide) P (Mutex) enfile (elt) V(SemaPlein) V(Mutex)	P(SemaPlein) P(Mutex) elt = defile () V(SemaVide) V(Mutex)

Les 2 autres versions entraînent une étreinte fatale :

- Dans la version 2, le consommateur ne sera pas bloqué sur une messagerie vide et à l'inverse le producteur ne sera pas bloqué sur une messagerie pleine : ceci entraînera une mauvaise utilisation du buffer.
- Dans la **version 3**, le producteur (fonction envoie) ne sera pas bloqué tant que le **buffer** ne sera pas plein ; quand ce sera le cas, le sémaphore **SemaVide** étant nul, il restera bloqué à l'intérieur de la section critique bloquant ainsi le consommateur (Retire) : c'est un inter-blocage.

Question D.2.5 Valeurs d'initialisation du mutex : 1

Question D.2.6 Méthode chargePages :

```

bool CCombinaison::chargePages (void)
{
    ifstream param; // fichier des débits unitaires
    Ligne ligne; // une ligne de débit
    int debit, nbLignes;
    string texteDebit, texteNbLignes;
    bool retcode = true ;
    table = new Page[nbPages];
    param.open(FICHIER_PAGES);

    while(!param.eof()) {
        param >> texteDebit >> debit >> texteNbLignes >> nbLignes;
        int noPage =debit/10;

        //***** Sortie méthode si fin de fichier inopinée *****
        if (param.eof () || !param.good ()) { //***** soit test flux
            retcode = false ;
            break ;
        }
        if (noPage < 0 || noPage >= nbPages) { //***** soit test numéro page
            retcode = false ;
            break ;
        }
        if (texteDebit != "DEBIT" || texteNbLignes != "LIGNES") {
            retcode = false ;
            break ;
        }
        for (int noLigne = 0; noLigne < nbLignes; noLigne++) {
            for (int noGroupe=0; noGroupe < NBGROUPES; noGroupe++)
                param >> ligne.debitGroupe[noGroupe];
            param >> ligne.puissance;
            table[noPage].push_back (ligne);
        }
    }
    param.close();
    return retcode;
}

```

Question D.2.7 Méthode combinaisonAdjacente:

```

int CCombinaison::combinaisonAdjacente(int noPage, Ligne ligneCourante) {
    for (unsigned int noLigne=0; noLigne < table[noPage].size(); noLigne++) {
        int nbAjoutsOuRetraits=0;
        for (int noGroupe=0; noGroupe < NBGROUPES; noGroupe++) {
            if ((ligneCourante.debitGroupe[noGroupe] != 0
                && this->table[noPage][noLigne].debitGroupe[noGroupe] == 0)
                || (ligneCourante.debitGroupe[noGroupe] == 0
                && this->table[noPage][noLigne].debitGroupe[noGroupe] != 0))
                nbAjoutsOuRetraits++;
        }
        if (nbAjoutsOuRetraits == 1) // c est une adjacente
            return noLigne;
    }
    return -1; // Pas de combinaison adjacente disponible
}

```


E.1.1 Dans la configuration présente le réseau Modbus peut-il supporter le nombre d'esclaves ? Justifiez votre réponse.

.Oui nombre d'esclaves < 31 (RS485 32 équipements max....

.....

.....

.....

.....

E.2.1 Classe d'adresse : C.....

Justification : Les adresses de classe C ont leurs 1^{ER} octet compris entre 192 et 223 (110X XXXX)

Masque réseau : 255.255.255.0

Adresse de diffusion : 192.168.1.255

E.2.2 Nombre d'équipements : 7

Nombre de bits pour l'adresse équipements : 4

Nombre de bits pour l'adresse sous-réseau : 4

Justification : Sur le réseau net1 ou net2, il y a 6 APIs et un routeur soit 7 équipements. 2 adresses sont réservées (réseau et broadcast) il faut 4 bits d'adresse pour les équipements, il en reste 4 pour les sous-réseaux.

E.2.3 Sous Réseaux

Il y a 8 possibilités d'adresse sous-réseau, on peut choisir n'importe lesquelles. Voici 1 exemple :

Nom	Adresse Réseau	Masque	Broadcast	Adresse mini	Adresse maxi
net1	192.168.1.0	255.255.255.224	192.168.1.31	192.168.1.1	192.168.1.30
net2	192.168.1.32	255.255.255.224	192.168.1.63	192.168.1.33	192.168.1.62
	192.168.1.64	255.255.255.224	192.168.1.95	192.168.1.65	192.168.1.94
	192.168.1.96	255.255.255.224	192.168.1.127	192.168.1.65	192.168.1.94
net3	192.168.1.128	255.255.255.224	192.168.1.159	192.168.1.129	192.168.1.158
	192.168.1.160	255.255.255.224	192.168.1.191	192.168.1.161	192.168.1.190
	192.168.1.192	255.255.255.224	192.168.1.223	192.168.1.193	192.168.1.222
	192.168.1.224	255.255.255.224	192.168.1.255	192.168.1.225	192.168.1.254

Remarque :

Les adresses réseaux 192.168.1.0 et 192.168.1.224 peuvent être utilisées d'après RFC bien qu'elles soient déconseillées dans certains livres, du fait que tous les bits de la partie sous-réseau sont à 0 ou à 1.

Equipement	Adresse IP	Equipement	Adresse IP
Router : eth0 eth1 eth2	192.168.1.30 (dans net1)		
	192.168.1.62 (dans net2)		
	192.168.1.158 (dans net3)		
PC supervision	192.168.1.129 (dans net3)		
BULL SPS 5	192.168.1.130 (dans net3)		
API-1	192.168.1.1 (dans net1)	API-2	192.168.1.33 (dans net2)
API-3	192.168.1.2 (dans net1)	API-4	192.168.1.34 (dans net2)
API-5	192.168.1.3 (dans net1)	API-6	192.168.1.35 (dans net2)
API-7	192.168.1.4 (dans net1)	API-8	192.168.1.36 (dans net2)
API-9	192.168.1.5 (dans net1)	API-10	192.168.1.37 (dans net2)
API-11	192.168.1.6 (dans net1)	API-12	192.168.1.38 (dans net2)

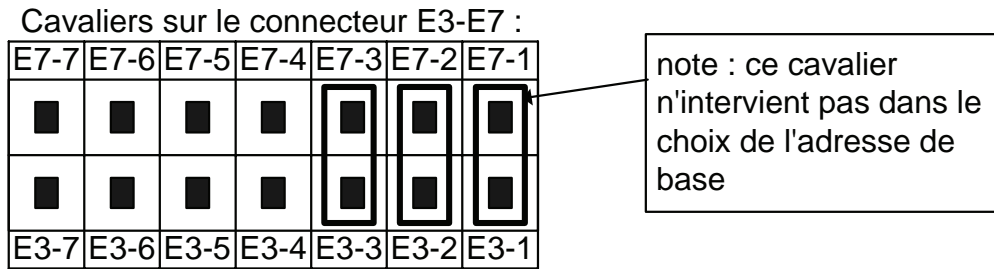
E.3.1 Remplir les champs contenus dans le tableau du document suivant :

CHAMP (IP)	VALEUR	CHAMP (TCP)	VALEUR
Version :	04	Port Source :	502
Type de service :	00	Port Destination :	1062
Identification :	40	Numéro d'ordre :	7EBC69D5
Durée de vie :	64	Numéro d'accusé de réception :	3E544F58
Protocole :	06	URG :	0
Somme de contrôle de l'en-tête :	1CCA	ACK :	1
Adresse source:	192.168.1.1	PSH :	1
Adresse destination :	192.168.1.129	FIN :	0
Nombre d'octets que comporte le datagramme IP	33	Somme de contrôle :	C9A2
		Nombre d'octets que comporte le datagramme TCP	20

E.3.2 Remplir les champs de la trame question (PC->API) :

CHAMP (MODBUS)	VALEUR	CHAMP (MODBUS)	VALEUR	CHAMP (MODBUS)	VALEUR
Identificateur de transaction	0000	Unit Identifier	00	Nombre de mots lus	01
Identificateur de protocole	0000	Code requête	03	Numéro du mot lu	0019
		Longueur	06		

F.1.1 Adresse de base de la carte VIPC610:



F.1.2 Adresse de base du module "IP-Serial-A" dans l'emplacement A :

\$F000.....

Adresse de base du module "IP-Serial-B" dans l'emplacement B :

\$F100.....

F.1.3 Adresse des registres :

"IP-Serial-A", Channel A, registre Control : **\$F005**

"IP-Serial-A", Channel A, registre Data : **\$F007**

"IP-Serial-A", Channel B, registre Control : **\$F001**

"IP-Serial-A", Channel B, registre Data : **\$F003**

F.1.4 Plage d'adresses de l'espace "SHORT I/O" du bus VME vue depuis la carte MVME147 : **\$FFFF0000 à \$FFFFFFFF**

F.2.1 Prédiviseur de fréquence : **1**

F.2.2 Time Constant TC = **$3\ 686\ 400 / (2 * 1 * 9600) - 2 = 190 = (0000\ 0000\ 1011\ 1110)$**

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
WR12	1	0	1	1	1	1	1	0
WR13	0	0	0	0	0	0	0	0

F.2.3 Registres WR4 et WR5 :

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
WR4	0	0	X	X	0	1	X	0
WR5	X	1	1	X	X	X	X	X