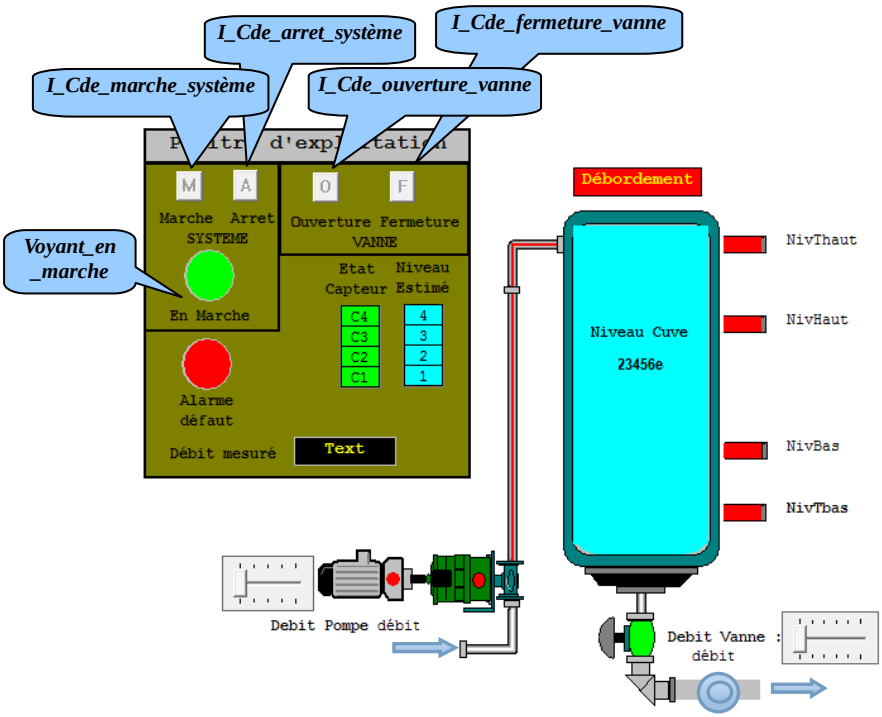
BTS MSP	Programmation sous Unity en FBD (langage ISO 61131).	7- Langage FBD
TD 3h		

I) présentation du système:

On souhaite automatiser le maintient d'un niveau dans une cuve. Le remplissage se fait par une motopompe de débit 2 l/s et la vidange par une vanne avec un débit de 3 l/s.

Schéma synoptique du système:



Cahier des charges n°1:

Le bouton poussoir M (***I_Cde_marche_système)*** met en service l'installation et le BP A (***I_Cde_arret_système)*** arrête l'installation. La mémorisation de la mise en service se fait par la variable ***KM_Mem_en_service***.

La pompe se met en marche si elle est en service et que le niveau descend en dessous du capteur Niveau Bas (I_NivBas). La pompe s'arrête si le niveau atteint le niveau très haut (I_NivThaut).

La vanne est commandée manuellement par les BP O (I_Cde_ouverture) et F (I_Cde_fermeture), mais ne peut pas s'ouvrir si le niveau est très bas (I_NivTbas).

Ce système sera simulé à l'aide du logiciel de programmation schneider UNITY pour automate industriels .

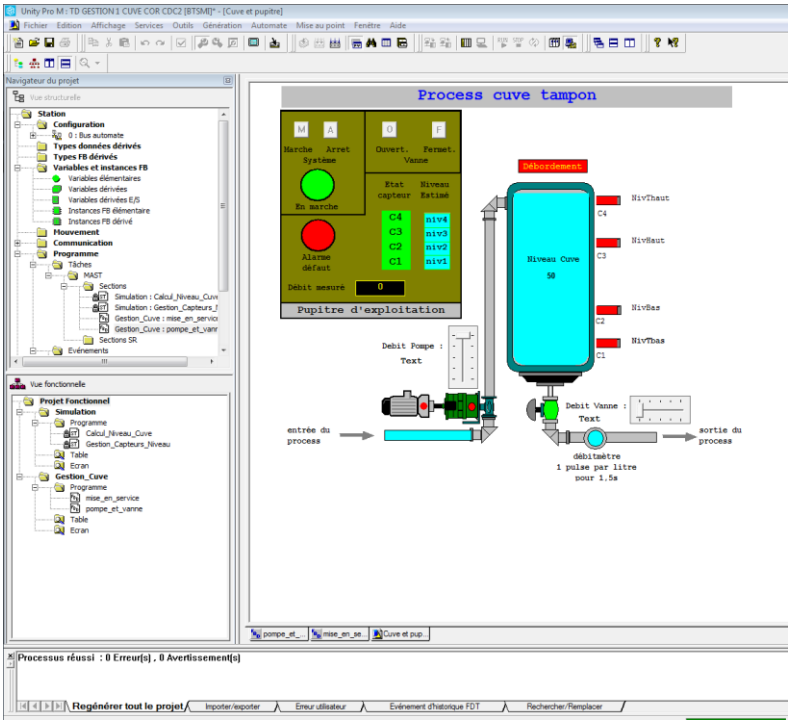
II) On vous demande de:

1. Découverte de l'interface d'Unity

-Lancer UNITY PRO V4

-Ouvrir le fichier "td gestion 1 cuve sujet [BTSMI]" ***(aller chercher le fichier sur NAS_BTSMI et le copier sur le disque local D:\votre nom\td gestion 1 cuve sujet [BTSMI])***

Vous devez obtenir la fenêtre suivante:



Dans la vue structurelle: (fenêtre de gauche en haut):

-Le menu « configuration » où on configure le matériel

-Le menu « variable élémentaires » liste les variables utiles pour ce projet.

-Le menu « programme » permet d'accéder aux différents modules d'un programme.

Dans la vue fonctionnelle: (fenêtre de gauche en bas):

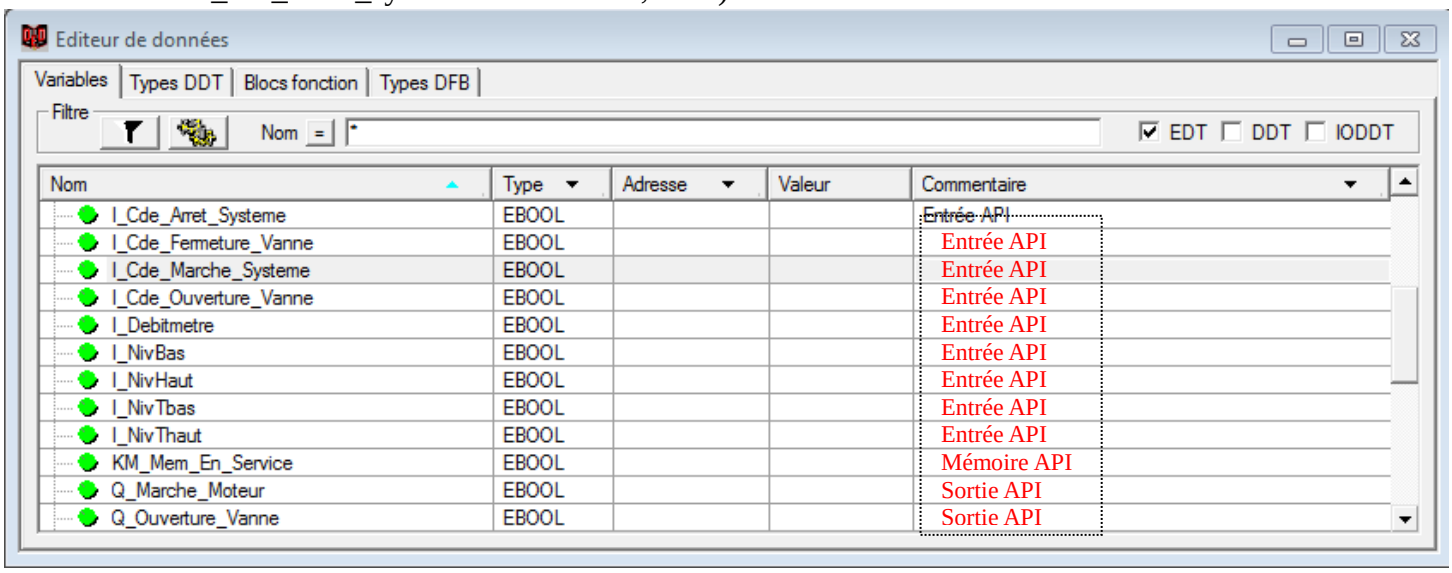
-La partie « simulation » contient la partie du programme (verrouillée pour vous) qui simule le comportement de la cuve , des capteurs …

-La partie "gestion" va contenir votre application de gestion du moteur de la pompe et de la vanne.

-La partie "0-Cuve et pupitre" contient l'écran d'exploitation: équivaut à un pupitre graphique XBT.

1.1. Naviguer dans les différentes parties afin de découvrir le logiciel...

1.2. Compléter dans le tableau des variables élémentaires la colonne Commentaire en expliquant à quoi correspond chaque mnémonique(Entrée API, Sortie API ou Mémoire API. (Ex: I_cde_Arret_systeme : Entrée API, etc...)



Nom	Type	Adresse	Valeur	Commentaire
I_Cde_Arret_Systeme	EBOOL			Entrée API
I_Cde_Fermeture_Vanne	EBOOL			Entrée API
I_Cde_Marche_Systeme	EBOOL			Entrée API
I_Cde_Ouverture_Vanne	EBOOL			Entrée API
I_Debitmetre	EBOOL			Entrée API
I_NivBas	EBOOL			Entrée API
I_NivHaut	EBOOL			Entrée API
I_NivTbas	EBOOL			Entrée API
I_NivThaut	EBOOL			Entrée API
KM_Mem_En_Service	EBOOL			Mémoire API
Q_Marche_Moteur	EBOOL			Sortie API
Q_Ouverture_Vanne	EBOOL			Sortie API

2. Mise en équation (logiques) du fonctionnement

1. donner l'équation de mise en service puis du Voyant marche de l'HMI :

   KM_Mem_en_service =/I_cde_arret_systeme.(I_cde_marche_systeme+KM_Mem_en_service)

   HMI_Voyant marche=KM_Mem_en_service

2. donner l'équation de la mise en marche du moteur:

   Set(Q_Marche_Moteur) =/I_NivBas.KM_mem_en_service

3. puis l'équation de mise à l'arrêt:

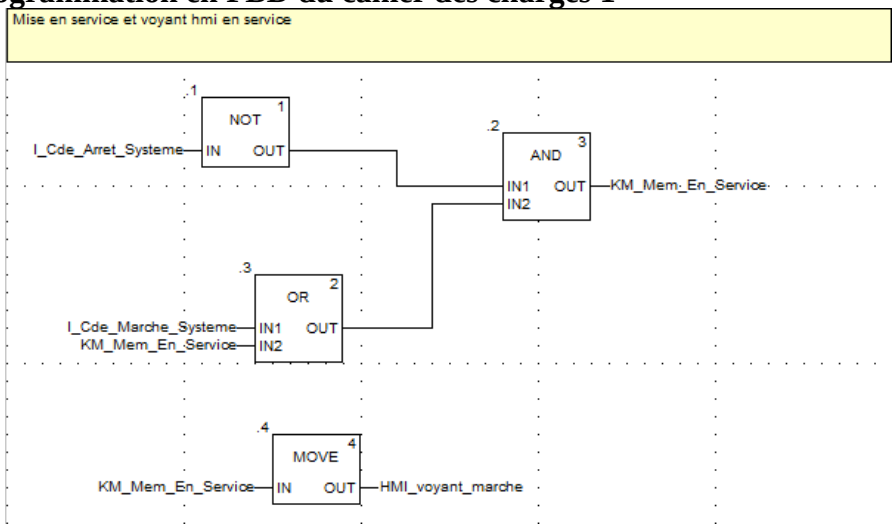
   Reset(Q_Marche_Moteur) =I_NivTHaut+/KM_mem_en_service

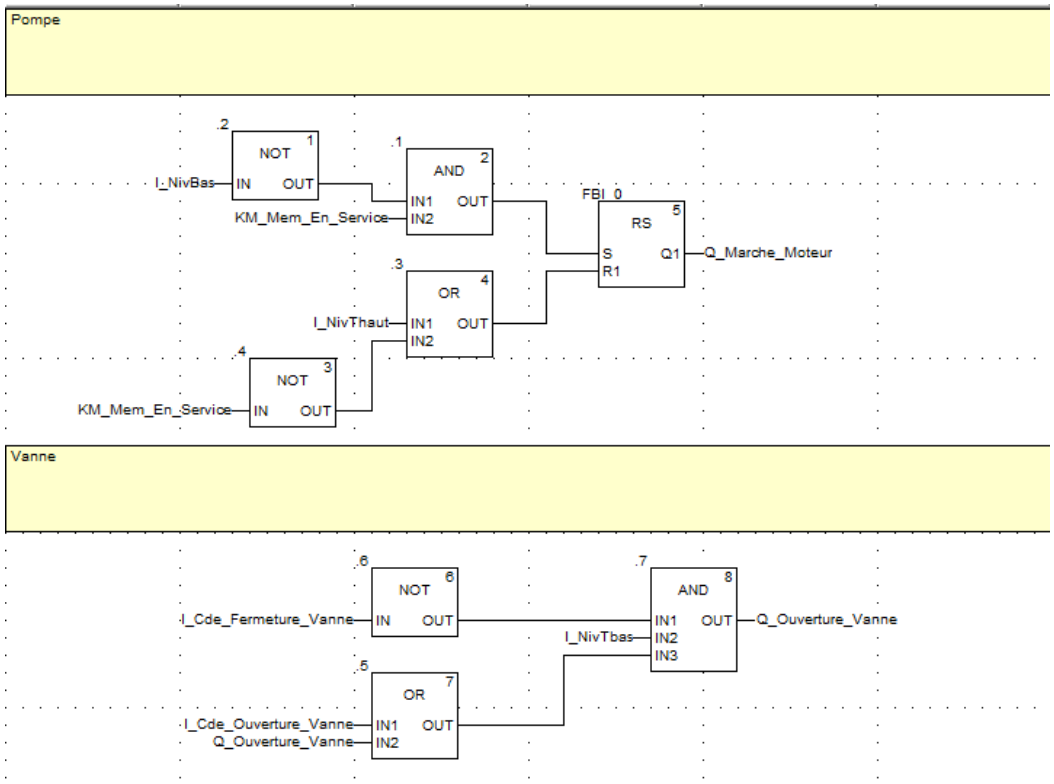
4. Enfin donner l'équation de l'ouverture de la vanne:

   Q_Ouverture_Vanne

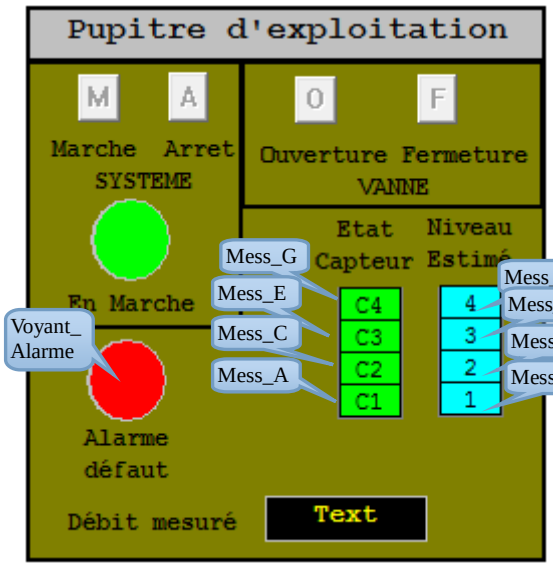
   =I_NivTbas./I_cde_fermeture_vanne.(I_cde_ouverture_vanne+Q_ouverture_vanne)

3. Programmation en FBD du cahier des charges 1





4. Cahier des charges n°2



On appelle C4 le capteur NivThaut, C3 le capteur Nivhaut, C2 le capteur NivBas et C1 le capteur NivTBas.
On désire afficher les informations suivantes :
Les voyants verts états des capteurs C1,C2,..C4 doivent représenter l'état des capteurs C1 à C4. Les voyants bleus niveau estimé 1,2,3 et 4 représentent le niveau estimé.
Le message « ALARME » doit s'allumer au cas où il y a discordance entre les informations en provenance des capteurs afin de signaler une défaillance possible d'un des capteurs. (Par exemple C3 à 1 alors que C1 est à 0: dans ce cas on affiche le niveau correspondant au capteur actif le plus haut, tous les niveaux estimé en dessous devront alors être affichés 1,2 et 3, par contre C1 ne sera pas affiché car le capteur pose problème).
On souhaite réaliser le programme en logigramme (bloc FBD) sur l' API M340.

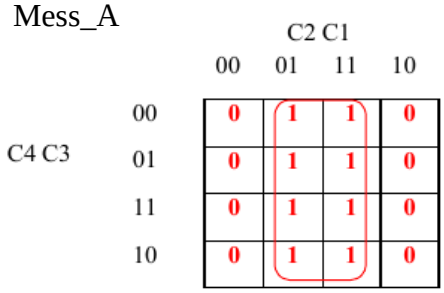
1. Faire une table de vérité représentant les sorties en fonction des combinaisons d'entrées C1, C2, C3 et C4 possibles.

Entrées				Sorties								
C4	C3	C2	C1	NIV1		NIV2		NIV3		NIV4		ALARME
				Mess_A	Mess_B	Mess_C	Mess_D	Mess_E	Mess_F	Mess_G	Mess_H	Voyant_alarme
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0	0	0	0
0	0	1	0	0	1	1	1	0	0	0	0	1
0	0	1	1	1	1	1	1	0	0	0	0	0
0	1	0	0	0	1	0	1	1	1	0	0	1
0	1	0	1	1	1	0	1	1	1	0	0	1
0	1	1	0	0	1	1	1	1	1	0	0	1
0	1	1	1	1	1	1	1	1	1	0	0	0
1	0	0	0	0	1	0	1	0	1	1	1	1
1	0	0	1	1	1	0	1	0	1	1	1	1
1	0	1	0	0	1	1	1	0	1	1	1	1
1	0	1	1	1	1	1	1	0	1	1	1	1
1	1	0	0	0	1	0	1	1	1	1	1	1
1	1	0	1	1	1	0	1	1	1	1	1	1
1	1	1	0	0	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	0

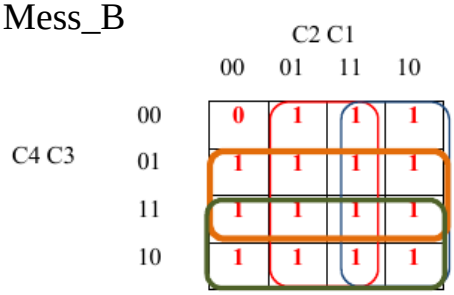
2. Établir les équations des 9 sorties en utilisant ci-besoin la méthode des tableaux de Karnaugh par exemple.

Remarque: les tableaux de karnaugh pour Mess_A,Mess_C,Mess_E et Mess_G ne sont pas utiles, on peut trouver directement leurs équations.

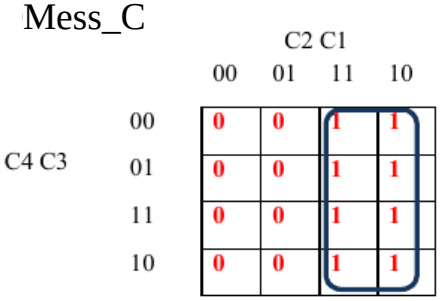
Mess_A



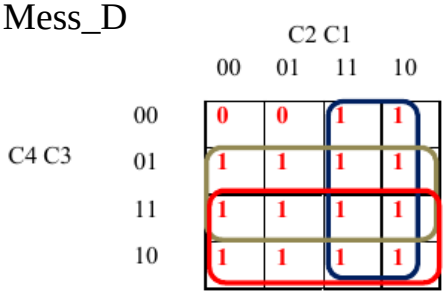
Mess_B



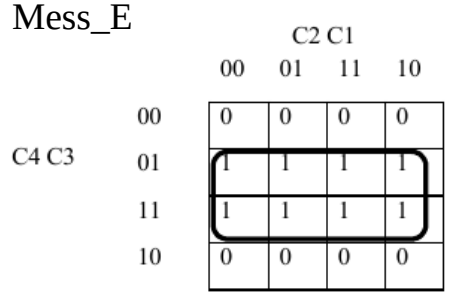
Mess_C



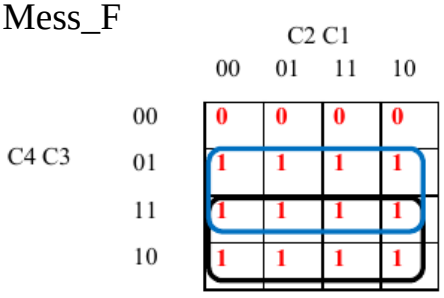
Mess_D



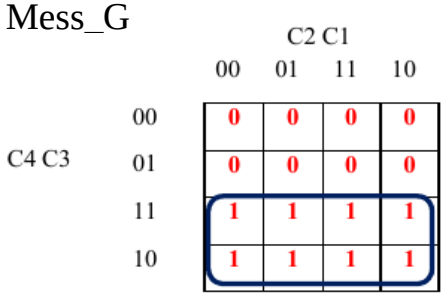
Mess_E



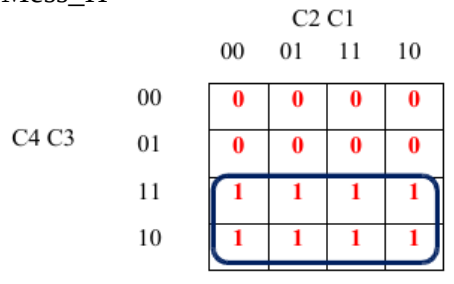
Mess_F



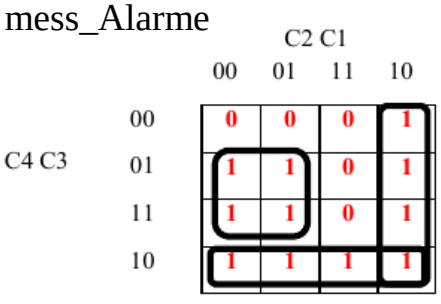
Mess_G



Mess_H



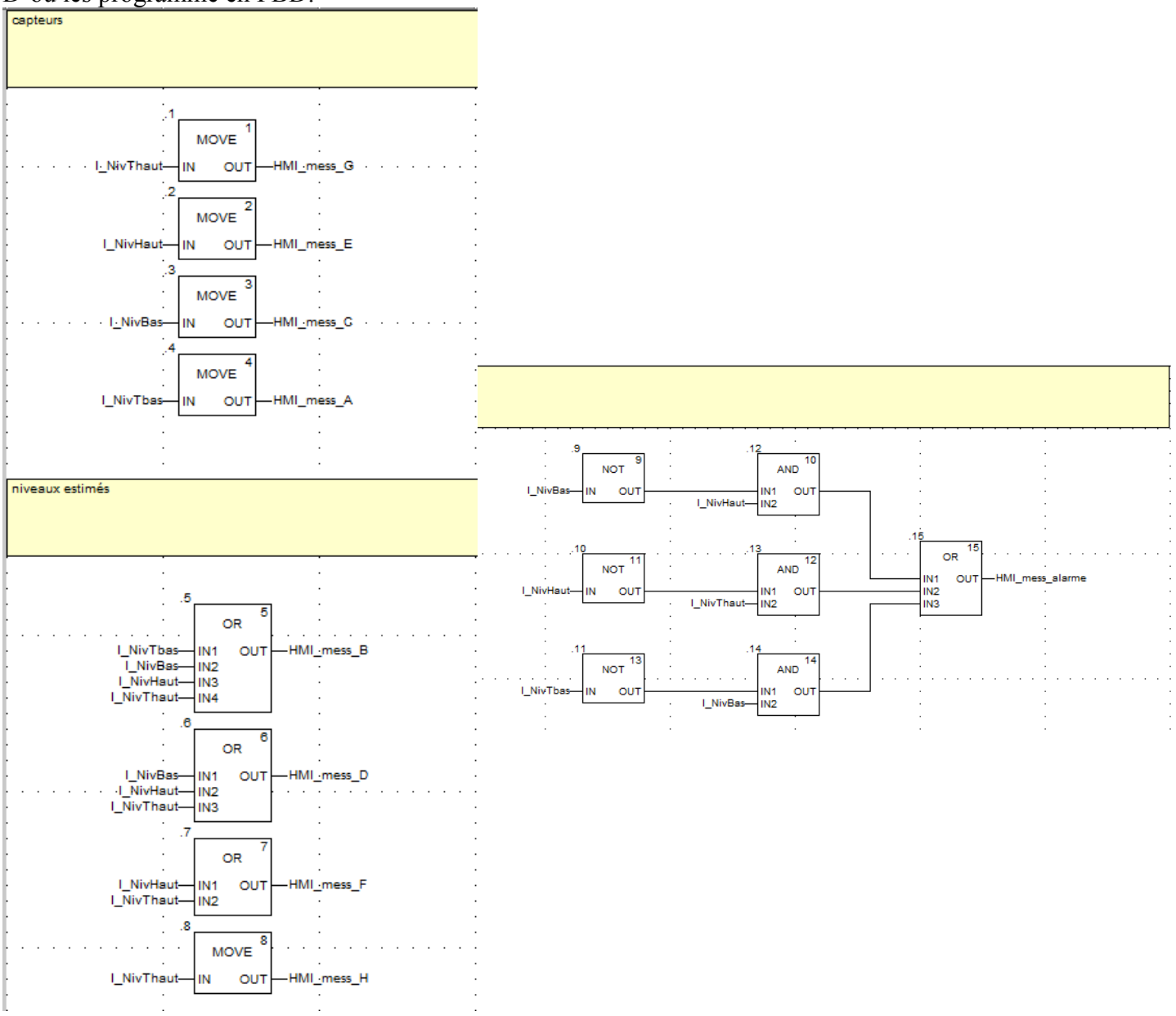
mess_Alarme



I_mess_A = I_NivTbas
I_mess_B = I_NivTbas+I_Nivbas+I_Nivhaut+I_NivThaut
I_mess_C = I_NivBas
I_mess_D = I_Nivbas+I_Nivhaut+I_NivThaut
I_mess_E = I_NivHaut
I_mess_F = I_Nivhaut+I_NivThaut
I_mess_G = I_NivTHaut
I_mess_H = I_NivTHaut
mess_alarme = I_NivBas./I_NivTbas+/I_NivBas.I_NivHaut+I_NivTHaut./I_NivHaut

Avec C1=I_NivTbas, C2=I_NivBas, C3=I_NivHaut, C4=I_NivTHaut

D'où les programme en FBD:



5. Cahier des charges n°3

On désire connaître le débit au niveau de la sortie et l'afficher en litres/minute. Pour cela, on dispose d'un débitmètre placé au niveau de la sortie de la vanne. Ce débitmètre délivre des impulsions ayant une fréquence proportionnelle au débit : f = 1 impulsion par litre/1,5s . (effectuer le comptage du nombre d'impulsions sur 3s afin d'avoir moins d'incertitude sur l'affichage)

Aide: On peut créer une base de temps de 1s avec le bit système %S6 ou de 100ms avec le bit %S5

Voici le programme avec une base de temps de 1,5s au lieu de 3s: