

SUPERVISION

Prise en main panorama et Kepserver :

Q1. Regarder la vidéo1, et faire ci-dessous le schéma réseau de notre supervision.

Q2. Regarder la vidéo2

Validation des différents	
---------------------------	--

. Regarder la vidéo3. Et expliquer clairement la problématique du pilotage avec retour d'état.

Q3. Regarder la vidéo4

Validation	
------------	--

- Q4. Regarder la vidéo 5.
Expliquer clairement l'erreur qui a été commise dans le programme.

Reporter ci-dessous un exemple de combinatoire de fin de cycle pour une sortie :

Mettre en œuvre un pilotage en mode manuel (Commande et retour d'état)

Validation	
------------	--

- Q5. Regarder la vidéo 5.

Validation	
------------	--

- Q6. Regarder la vidéo 6

Validation	
------------	--

- Q7. Réaliser un composant pupitre de supervision du GC propre et complet

Validation	
------------	--

- Q8. Réaliser un composant animation vérin :

Validation	
------------	--

Réalisation d'un bloc fonction

On peut imaginer que dans les projets d'automatisme une entreprise est souvent à piloter des vérins. Pour gagner du temps en ne pas avoir à redémarrer de zéro à chaque fois, on vous propose d'étudier la réalisation de bloc fonction.

Vous trouvez ci-dessous le bloc fonction appeler Surv_Verin :

Description du bloc fonction de surveillance des vérins :

Unity Pro XL: <No name> - [Data Editor]

File Edit View Services Tools Build

Project Browser

Structural view

Project

- Configuration
 - 0 : PLC bus
- Derived Data Types
 - Derived FB Types
 - FB_surv_verin
 - Sections
 - ST Prog_surv_verin
 - LD Temperreux
- Variables & FB instances
 - Elementary Variables
 - Derived Variables
 - Device DDT Variables
 - IO Derived Variables
 - Elementary FB Instances
 - Derived FB Instances
- Motion
 - Derived FB Instances
- Communication
 - Networks
- Program
 - Tasks
 - MAST
 - Sections
 - LD verin
 - CD Sections

Variables DDT Types Function Blocks DFB Types				
Filter				
Name	no.	Type	Value	C
FB_surv_verin		<DFB>		
<inputs>				
CP_verin_rentre	1	BOOL		
CP_verin_sortie	2	BOOL		
Commande_Sortir_verin	3	BOOL		
Commande_rentre_verin	4	BOOL		
temps_erreur	5	TIME		
<outputs>				
Default	1	BOOL		
Nb_cycle_complet	2	INT		
Mot_etat	3	WORD		
<inputs/outputs>				
<public>				
Etat_verin		INT		
<private>				
<sections>				
ST Prog_surv_verin		<ST>		
LD Temperreux		<LD>		

Programme en ST appelé : « Prog_surv_verin »

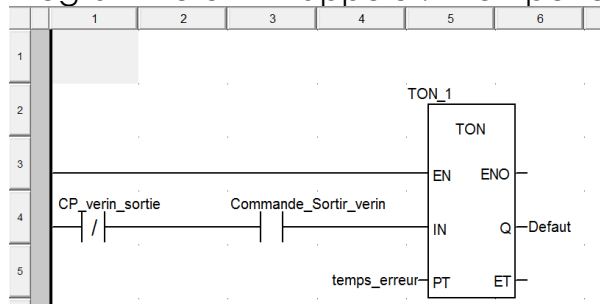
```

IF (Etat_verin=0 AND Commande_Sortir_verin) THEN Etat_verin:=1;
END_IF ;
IF (Etat_verin=1 AND CP_verin_sortie ) THEN Etat_verin:=2;
END_IF ;
IF (Etat_verin=2 AND Commande_rentre_verin) THEN Etat_verin:=3;
END_IF ;
IF (Etat_verin=3 AND CP_verin_rentre ) THEN
Etat_verin:=0;
Nb_cycle_complet:=Nb_cycle_complet+1;
END_IF ;

Mot_etat.0 := CP_verin_rentre;
Mot_etat.1 := CP_verin_sortie;
Mot_etat.2 := Commande_rentre_verin;
Mot_etat.3 := Commande_Sortir_verin;
Mot_etat.4 := Default;

```

Programme en LD appelé : « Temperreux » :



Q9. Expliquer le cahier des charges qui a permis de définir ce bloc fonction :

Q10. Lister les entrées et préciser leur signification.

Q11. Lister les sorties et préciser leur utilité.

Q12. Vous pouvez coder tester et valider le fonctionnement, ou récupérer le bloc fonction déjà codé en faisant une importation de « FB_SUP_Verin.XDB » .

Q13. Expliquer clairement l'utilité des bloc fonction.

Q14. Réaliser un composant de supervision Vérin conforme à ce qui a été programmé dans le bloc fonction :

Animation du vérin	
Apparition des boutons pilotage manu qu'en mode manu	
Voyant alerte erreur	
Affichage Nb cycle	
Control du temps d'erreur	
Afficher un voyant maintenance si le cycle le nombre de cycle dépasse une certaine limite.	