|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | Travaux pratiques :  **Programmer avec Automation Studio**  « Codage et visualisation IHM VC4 »  (Durée : 8h) | | Nom : - ……………………………………..  Classe :  **BTS CIEL IR**  Date :  **/ /** | | | | | |
| **Objectif séquence :** Réaliser le programme avec l’outil de développement Automation Studio de B&R | | | | | | | | | |
| **On donne :**   |  |  | | --- | --- | | 1. Une documentation sujet 2. Une documentation du constructeur avec des **Tutoriels BR et des TM BR** 3. Une documentation école avec des **Tutoriels** **Professeur** 4. Un logiciel « automation studio 4.9 minimum » 5. Un site internet B&R 6. Un PC de développement |  | | | | | | | | | | |
| **Pré Requis :** Outil de développement « Automation Studio 4» | | | | | | | | | |
| **Activité : D2** / Développement Logiciel | | | | | | | | | |
| **EVALUATION DES COMPETENCES VISEES :** | | |  | A : Acquis | E : Non acquis | | | | |
| **Compétences** | **Objectifs** | | | | **Niveau** | | | | |
| **A** | **B** | **C** | **D** | **E** |
| **C8** : Coder | Créer un programme de commande avec les langages LD et ST | | | |  |  |  |  |  |
| **C8** : Coder | Créer un programme de supervision IHM | | | |  |  |  |  |  |
|  |  | | | |  |  |  |  |  |
| **Note :** | | | | | / 20 | | | | |

***Vous compléterez et rendrez en fin de séance, le questionnaire papier ci-dessous***

***Vous serez évalué sur la clarté de votre compte rendu.***

Contents

[1. Contexte 2](#_Toc159430139)

[2. Créer un projet et choisir les composants du système 2](#_Toc159430140)

[3. Réalisation des programmes de commande avec le langage LaDder (LD) 3](#_Toc159430141)

[4. Réalisation d’une supervision IHM des programmes de commande LD 7](#_Toc159430142)

[5. Réalisation des programmes de commande avec le langage Structured Text (ST) 8](#_Toc159430143)

[6. Réalisation d’une supervision IHM des programmes de commandes ST 8](#_Toc159430144)

[7. Synthèse TP : 9](#_Toc159430145)

[8. Pour aller plus loin 10](#_Toc159430146)

[9. Annexes : 11](#_Toc159430147)

[1) Ajout d’un programme de comptage décimal en langage ST 11](#_Toc159430148)

[2) Liaison PC / CPU 13](#_Toc159430149)

[3) Transfert vers la cible CPU 14](#_Toc159430150)

# Contexte

A l’aide des tutoriels mis à votre disposition et du document TP ci-dessous, vous allez appliquer les différentes procédures pour réaliser différents programmes de commande en mode simulation avec les langages de programmation LADDER et STRUCTURED TEXT. Puis vous testerez vos programmes avec la réalisation d’une supervision IHM.

# Créer un projet et choisir les composants du système

**Etape1 : Créer un projet avec le logiciel Automation Studio et la CPU PPc70**

1. A l’aide du tutoriel 1, lancer le logiciel **Automation Studio**sur votre PC : (version logicielle : **V\_4**.9 minimum),

Préciser ci-contre la version Automation Studio de votre PC de développement : ……………….

1. Créer un nouveau projet et l’enregistrer sous la forme :

***TprojectNOMxxxxxx*** avec **NOM** = votre nom de famille et la date du jour **xxxxxx** avec dans l’ordre:

**xx** : année / **xx** : mois / **xx** : jour

1. Définir le nom de la configuration : **P70CPU  *et non celle du tutoriel : X20CPU***



1. Choisir la CPU**: 4PPC70.0702.20B *et non celle du tutoriel : X20CP1586***



1. Page 7/7 du tutoriel 1, cocher **🗹 Activate Simulation**
2. Vérifier dans la barre des tâches la présence du feux tricolore **ARsim C4.90 RUN,**

*(dans le cas contraire assurez vous que l’onglet feux tricolore dans le bandeau supérieur d’AS est activé)*

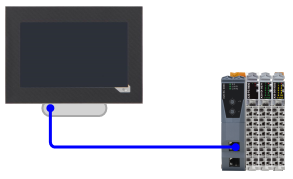


|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tâche** | **A** | **B** | **C** | **D** | **E** | **Signature** |
| Désignation du projet |  |  |  |  |  |  |
| Config Name |  |  |  |  |  |  |
| CPU |  |  |  |  |  |  |
| RUN en mode simulé |  |  |  |  |  |  |

**Etape2 : Réaliser l’implantation du matériel**

1. A l’aide du tutoriel 2, réaliser l’implantation des matériels ci-dessous dans le logiciel de développement Automation Studio, avec une liaison PowerLinK (PLK) et respectant le synoptique de la maquette système ci-dessous.

* **X20BC0083** : pour la liaison de la CPU « PPC70 » aux interfaces Entrées/Sorties
* **X20PS9400**: pour l’alimentation des modules X20 en 24V continu
* **X20DI9371** : interface d’entrées pour les capteurs
* **X20DO9322** : interface de sorties pour les actionneurs



Synoptique de la maquette :

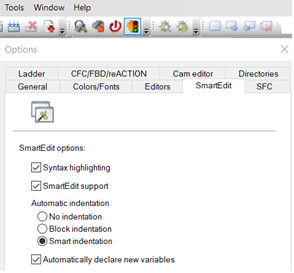
1. Relier le module X20BC0083 à votre CPU PPc70 avec une liaison Powerlink (PLK)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tâche** | **A** | **B** | **C** | **D** | **E** | **Signature** |
| Modules X20 |  |  |  |  |  |  |
| Liaison EPL |  |  |  |  |  |  |

# Réalisation des programmes de commande avec le langage LaDder (LD)

**Etape3 : Ajouter un programme en langage « Ladder »**

A l’aide du tutoriel 3 et du TM 240, réaliser le programme « LampTest » pour simuler le fonctionnement d’une lampe avec son interrupteur



1. Effectuer les réglages ci-dessous dans AS

* Procédure pour une déclaration « automatique » des variables*:*

⮱ **Tools / option / smart Edit** et cocher :

* Procédure pour Modifier le bloc fonction mapp / Ladder, :

⮱ **Tools / options / ladder** : et décocher

❑ Use EN/EN0 for function and function blocks

❑ Enable Synchronous Powerflow

1. Dans AS, les programmes sont écrits sous l’onglet **Logical View :**

**⮱** Click **Logical View** puis suivre la procédure du tutoriel 3

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tâche** | **A** | **B** | **C** | **D** | **E** | **Signature** |
| Programme Name : LampTest |  |  |  |  |  |  |

1. Créer les variables du programme

⮱ A l’aide du tutoriel 4, créer les variables :

Lamp : variable représentant l’actionneur « Lampe »

gSwitch : variable représentant le capteur « Interrupteur »

1. Définir les raisons d’avoir des variables locales ou globales :
2. Programmation Ladder : Suivre le tutoriel 5, vous reporterez ci-dessous votre programme commenté avec le langage LADDER

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tâche** | **A** | **B** | **C** | **D** | **E** | **Signature** |
| Variable locale : Lamp |  |  |  |  |  |  |
| Variable globale : gSwitch |  |  |  |  |  |  |
| Programme Ladder : PROGRAM CYCLIC |  |  |  |  |  |  |

**Etape 5 : Compiler votre programme et transférer votre projet dans votre CPU simulé**

1. A l’aide du tutoriel 7, Compiler et transférer votre programme

**Etape 6 : Vérifier le bon fonctionnement de votre programme**

1. A l’aide du tutoriel 8, Simuler le fonctionnement de votre programme

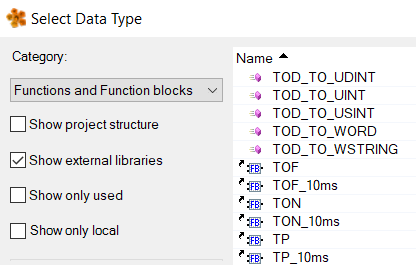
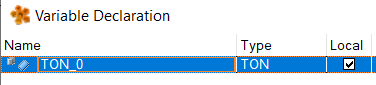
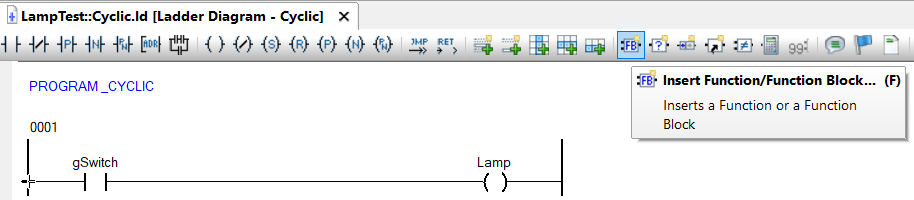
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tâche** | **A** | **B** | **C** | **D** | **E** | **Signature** |
| Test du programme « LampTest » en simulation |  |  |  |  |  |  |

**Etape 7 : Exercices de codage avec le langage LADDER**

1. A l’aide du TM240, réaliser et tester les 7 programmes des pages 31/44 et 32/44. Vous préciserez alors, ci-dessous, la fonction de chacun :

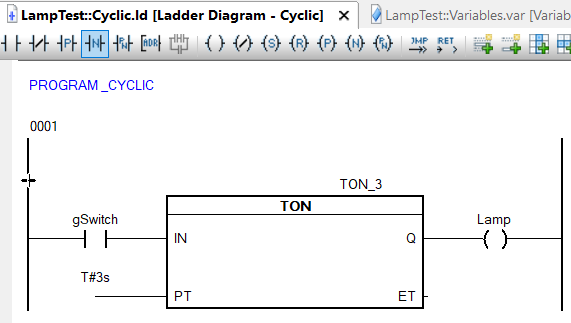
* Programme 1 :
* Programme 2 :
* Programme 3 :
* Programme 4 :
* Programme 5 :
* Programme 6 :
* Programme 7 :

1. A l’aide du TM240, page 33, réaliser et tester l’ajout d’un Block Fonction TON à votre programme Ladder « LampTest », vous préciserez alors, ci-dessous le rôle de ce bloc fonction



Vous préciserez sur votre bloc fonction :

* Nom de la TON : **TON\_3**
* PT = **3** secondes

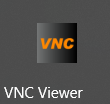
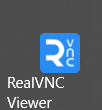


|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tâche** | **A** | **B** | **C** | **D** | **E** | **Signature** |
| Test du programme en simulation avec la TON |  |  |  |  |  |  |

# Réalisation d’une supervision IHM des programmes de commande LD

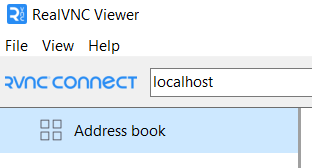
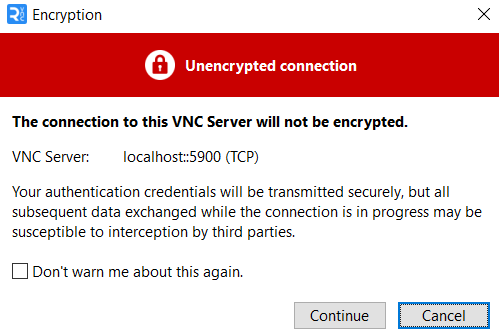
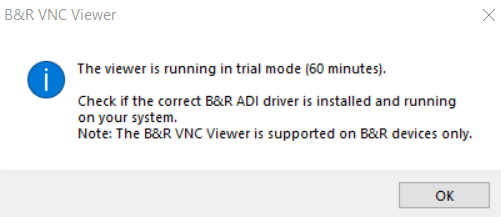
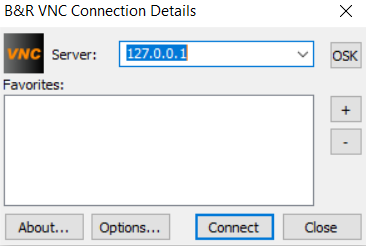
**Etape 1 : Visualisation des résultats avec une IHM**

A l’aide de votre logiciel de développement Automation Studio, vous allez créer une visualisation IHM de votre programme « **LampTest** » obtenu à la question précédente et tester le résultat sur un écran de PC grâce à l’outil logiciel « VNCviewer ».



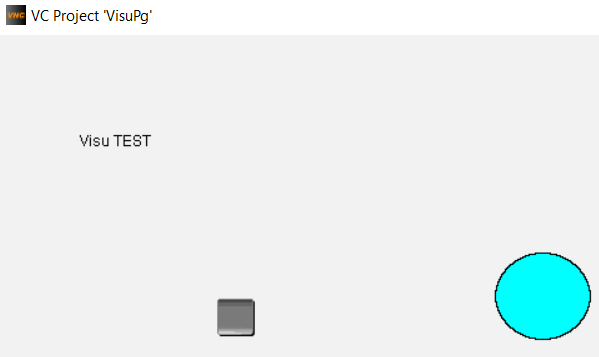
1. Télécharger l’outil logiciel VNCviewer :

ou



NB : le VNC server est géré par Automation Studio

1. A l’aide du tutoriel « TutoBRsoft\_ASvisuIHMvc4\_23», ajouter un titre à la visu, un bouton et un voyant

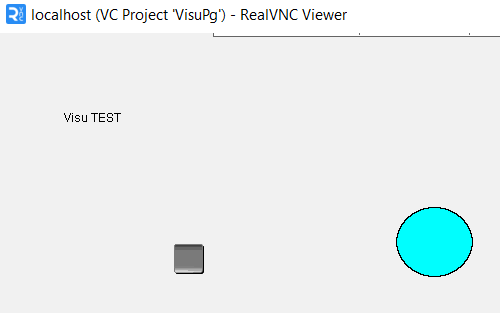


Titre : **VisuTEST**

Bouton : **BPonoff**

Voyant : Cercle de couleurs (2 couleurs de votre choix)

1. Tester le fonctionnement de votre IHM avec un changement de couleur du voyant après chaque appui touche sur le BP\_ONOFF



Vous préciserez ci-dessous le temps observé pour l’allumage du voyant :

……………………………………

Vous préciserez ci-dessous le temps observé pour l’extinction du voyant :

…………………………………….

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tâche** | **A** | **B** | **C** | **D** | **E** | **Signature** |
| Visualisation IHM |  |  |  |  |  |  |
| Test IHM « LampTest » |  |  |  |  |  |  |

# Réalisation des programmes de commande avec le langage Structured Text (ST)

1. A l’aide du TM246 et des annexes du TP, **réaliser et tester** un programme de comptage « PgCPT » avec le langage ST

Cahier des charges : Programme de commande

* Langage ST
* Création d’une Variable de comptage **CPT** qui évolue de 0 à 255

Cahier des charges : Programme de supervision IHM

* Création d’un Bouton StarT/ StoP qui lance le comptage puis l’arrête : **BP\_StSp**
* Création d’un Bouton de remise à zéro du compteur : **BP\_RAZ**

1. Ecrire votre programme ST commenté ci-dessous :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tâche** | **A** | **B** | **C** | **D** | **E** | **Signature** |
| Test du Programme « PgCPT » |  |  |  |  |  |  |

# Réalisation d’une supervision IHM des programmes de commandes ST

1. A l’aide du tutoriel « TutoBRsoft\_ASvisuIHMvc4\_23» et de l’outil logiciel VNC, ajouter un titre à la visu, 2 boutons BP\_StSp , BP\_RAZ et un champ numérique nous permettant de voir le comptage évolué.

Titre : Compteur

Boutons : BP\_StSp et PB\_RAZ

Champ du comptage :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tâche** | **A** | **B** | **C** | **D** | **E** | **Signature** |
| Test IHM « PgCPT » |  |  |  |  |  |  |

|  |  |
| --- | --- |
| Synthèse TP : | ***AS : Control et Visualisation IHM (VC4)*** |

1. A l’aide de la documentation numérique mise à votre disposition : « TM210TRE.30 », relever la signification des onglets :

**Logical View** :

**Configuration View :**

**Physical View** :

1. A l’aide de la documentation numérique mise à votre disposition relever ci-dessous les langages de programmation utilisés :
2. Préciser ci-dessous les caractéristiques du langage de programmation LADDER, en déduire la, les raison(s) de son utilisation
3. Préciser ci-dessous les caractéristiques du langage de programmation STRUCTURED TEXT en déduire la, les raison(s) de son utilisation
4. Ces langages appartiennent-ils à la norme IEC61131-3, si oui préciser l’intérêt de la norme :

# Pour aller plus loin

**Etape n : mapper les Entrées/Sorties physiques aux variables du programme**

1. A l’aide du tutoriel 6, associer les variables du programme aux Entrées/Sorties physiques de la maquette
2. Définir ci-dessous le rôle du mapping en programmation
3. Simuler votre programme « LampTest » avec le watch d’AS, que remarquer vous ?

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tâche** | **A** | **B** | **C** | **D** | **E** | **Signature** |
| Test Programme avec mapping |  |  |  |  |  |  |

**Etape n : ajouter sur votre visu IHM VC4,**

1. L’heure et la date du jour, vous rédigerez ci-dessous la procédure utilisée sous la forme d’un mode d’emploi

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tâche** | **A** | **B** | **C** | **D** | **E** | **Signature** |
| IHM : Heure / date |  |  |  |  |  |  |

1. Le logo « Campus » de St Félix La Salle, vous rédigerez ci-dessous la procédure utilisée sous la forme d’un mode d’emploi

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tâche** | **A** | **B** | **C** | **D** | **E** | **Signature** |
| IHM : Image |  |  |  |  |  |  |

**Etape n : Transférer votre programme « IHM VC4 » dans une CPU type PPc70,**

A l’aide de la procédure mise à votre disposition, tester votre IHM à l’aide de l’écran tactile de la CPU « PPc70 »

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tâche** | **A** | **B** | **C** | **D** | **E** | **Signature** |
| Test IHM\_VC4 sur PPc70 |  |  |  |  |  |  |