|  |  |
| --- | --- |
| ***Étude des systèmes Discrets***  ***Analyser, Modéliser, Résoudre*** | |
| *Cycle 12* | ***Rotation de TP*** |
| *Supports* | |  |  | | --- | --- | | MAXPID |  | | ***Maxpid*** | ***Velo à assistance électrique*** | |  |  | | ***Commande d’un moteur pas à pas*** | ***Pilote hydraulique de bateau*** | |
| *Objectifs* | * ***Réaliser la partie commande d’un composant programmable*** |
| *Documents* |  |
| *Pré requis* |  |

# Objectifs

## Contexte pédagogique

|  |  |
| --- | --- |
|  | **Analyser :**   * A1 – Identifier le besoin et définir les exigences du système * A2 – Définir les frontières de l'analyse   **Modéliser :**   * Mod1 – Justifier ou choisir les grandeurs nécessaires à la modélisation * Mod2 – Proposer un modèle * Mod3 – Valider un modèle   **Expérimenter :**   * Exp2 – Justifier et/ou proposer un protocole expérimental   **Communiquer :**   * Com2 – Mettre en œuvre une communication |

## Évaluation des écarts

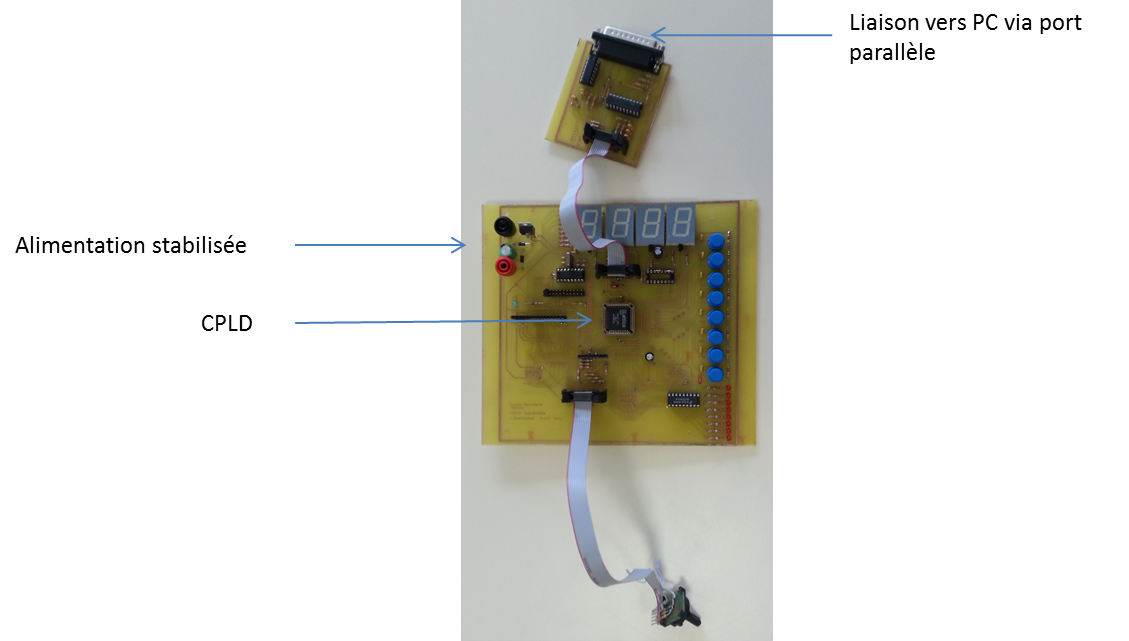
|  |  |
| --- | --- |
|  | **Au cours de ce TP on prendra garde à évaluer chacun des écarts.** |

## Problématique

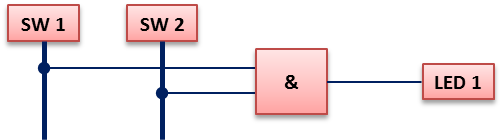
|  |
| --- |
| **Programmer un FPGA.** |

# Prise en main d’une carte programmable

L’objectif de cette partie est de prendre en main une carte programmable. Pour cela, on utiliser un CPLD (*Complex Programmable LogicDevice* – Circuit logique programmable).



On présente ici une méthode permettant d’allumer la LED 1 en appuyant simultanément sur les boutons poussoirs SW 1 et SW2 :



## Connexion de la carte et lancement

* Commencer par alimenter la carte à l’aide de l’alimentation stabilisée en 7V.
* Connecter la carte au port parallèle de l’ordinateur.
* Démarrer l’ordinateur et connectez-vous en utilisant le compte Eleve/Eleve.
* Sur le Bureau, aller dans le dossier PTSI :
  + Faites un copier-coller du dossier PTSI.
  + Renommer le répertoire en utilisant vos noms.
* Lancer le logiciel ispLEVER Classic Project Navigator :
  + Menu Démarrer ;
  + ispLEVER Classic Project Navigator ;
  + ouvrir le fichier DecouverteISP.
* Le fichier découverte a été préalablement configuré pour utiliser le CPLD présent sur la carte.

### Réalisation du logigramme

* Ouvrir le schéma « testled ».
* Si la boîte Drawing n’est pas ouverte :
  + View
  + DrawingToolbar
* Ajouter les portes d’entrée et de sortie :
  + Drawing toolbar : Add Symbol
  + Generic/ipopads.lib
  + Ajouter une porte G\_INPUT
  + Ajouter une porte G\_OUTPUT
* Ajouter le symbole de la porte ET :
  + Generic/gates.lib/G\_2AND
* Ajouter des fils en amont et en aval de la porte
* Ajouter des « Net Name » :
  + Cliquer sur Add net Name
  + Saisir SW1
  + Taper entrée
  + Cliquer sur le bout du fil où SW1 est raccordé
* Faire de même pour SW2, et LED1
* Ajouter des I/O Marker
  + Cliquer Add I/O marker
  + Cliquer Input
  + Cliquer au bout des fils au niveau de SW1 et SW2
  + Cliquer sur Output
  + Cliquer au bout du fil au niveau de LED1
* Sauvegarder et fermer votre schéma
* Double cliquer sur Compile Schematic pour Compiler votre schéma

### Association des entrées sorties Systèmes aux Entrées sorties du CPLD

* Se déplacer sur ispLSI1016-60LH44/883
* Double cliquer sur Constraint Editor
* Cliquer sur l’icône PackageView si le schéma du CPLD ne s’affiche pas.
* Faire glisser les entrées et les sorties conformément aux schémas électriques simplifiés.
* Sauvegarder et fermer l’éditeur de contraintes.
* Générer le fichier JEDEC
  + Double cliquer sur JEDEC file

### Téléchargement du programme sur le CPLD

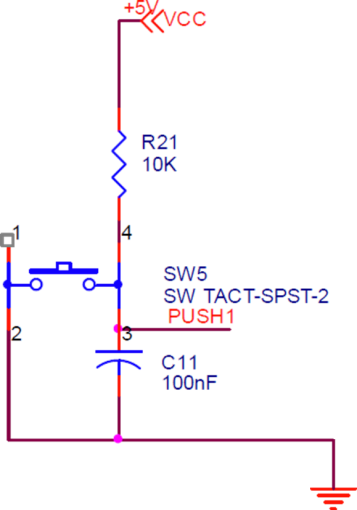
* Menu Tools
* ispVM System
* double cliquer le device et s’assurer que le bon DATA file est sélectionné.
* Cliquer sur GO

### Test du programme

* Débrancher le connecteur entre la grande et la petite carte.

1. La LED 1 est-elle allumée ou éteinte ? Que se passe-t-il quand on appuie sur les boutons SW1 et SW2 ?

On donne le schéma de câblage du bouton poussoir :



1. Expliquer son fonctionnement. Est-il normalement ouvert ? normalement fermé ? Est-ce compatible avec vos observations de la question précédente ?
2. Modifier le schéma pour répondre au cahier des charges initial.
3. Vérifier le bon fonctionnement de votre système.

*Remarque :* Pour complémenter un signal, il est possible d’utiliser un bloc « inverseur » (generic/gates.lib G\_INV).

# Présentation du fonctionnement des moteurs pas à pas

Vous allez analyser les modes de commande des moteurs pas à pas. Il existe deux types de machines : les moteurs unipolaires (4 phases) et les moteurs bipolaires (2 phases).

## Moteur unipolaire

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |
| --- | --- |
| Séquencement une phase à la fois | Séquencement deux phases à la fois |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | ***I11*** | ***I12*** | ***I21*** | ***I22*** | rotor | Pas | Sens de rotation | | I | 0 | 0 | 0 |  | 1 | | 0 | 0 | I | 0 |  | 2 | | 0 | I | 0 | 0 |  | 3 | | 0 | 0 | 0 | I |  | 4 | | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | ***I11*** | ***I12*** | ***I21*** | ***I22*** | rotor | Pas | | **I** | **0** | **I** | **0** |  | **1** | | **0** | **I** | **I** | **0** |  | **2** | | **0** | **I** | **0** | **I** |  | **3** | | **I** | **0** | **0** | **I** |  | **4** | |

Séquencement à demi pas

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | ***I11*** | ***I12*** | ***I21*** | ***I22*** | rotor | Pas | | **I** | **0** | **0** | **0** |  | **1** | |  |  |  |  |  | **2** | |  |  |  |  |  | **3** | |  |  |  |  |  | **4** | |  |  |  |  |  | **5** | |  |  |  |  |  | **6** | |  |  |  |  |  | **7** | |  |  |  |  |  | **8** | | Les flèches représentent les positions fictives du rotor.   1. À l’aide des séquencements précédents, indiquer quels sont les courants à faire circuler pour obtenir la position indiquée du rotor. 2. Tracer les chronogrammes des 4 phases pour un sens de rotation avec le séquencement « 1 phase à la fois ». |

# CARTE DE COMMANDE (minimaliste):

|  |
| --- |
| **Demander de l’aide au professeur pour brancher le GBF sur la carte.** |

1. À l’aide du logiciel ispVM programmez la carte avec le fichier « pas\_pas ». Si le moteur ne tourne pas appuyer sur le poussoir n°1 (SW1). Les led 1 à 4 doivent s’allumer

## Commande interne :

1. Relever les chronogrammes des quatre tensions de sorties aux bornes des shunts (la synchronisation sera prise sur la phase 1).
2. L’allure est-elle conforme à celle prévue précédemment ?
3. Indiquer entre quelles limites la fréquence (sans perte de pas ou décrochage) ?

# INVERSION DU SENS DE ROTATION :

## Identification des besoins

1. Tracez lez chronogrammes des signaux de commande du moteur pas à pas dans les deux sens.

## Détermination des équations :

On veut que le moteur tourne dans le sens inverse lorsqu’on appuie sur SW8

1. Donnez les équations des sorties PH1 à PH4 (commande du moteur) en fonction de l’entrée du sens de rotation (SW8) ainsi que des cycles S1 à S4 sachant que le sens horaire est défini vers la droite. Se référer à l’annexe.
2. Ouvrir le projet « pas\_pas.syn » et modifier le schématique à partir des équations trouvés.
3. Générer le fichier jedec et reprogrammer la carte

# Fonctions complémentaires

1. Proposer une solution pour arrêter la rotation via une entrée (SW7 par exemple).
2. Proposer une solution pour qu’un appui sur SW8 change le sens de rotation puis un nouvel appui réinverse le sens.

