Informatique industrielle TP

v1

IUT d'Annecy, 9 rue de l'Arc en Ciel, 74940 Annecy

Conversions Analogique-Numérique

1 Révisions

Cahier des charges : Entrées et sorties logiques

- La diode D0 est allumée si le bouton poussoir BP0 est appuyé et le bouton poussoir BP1 est relâché.
- La diode D1 est allumée si le bouton poussoir BP0 est relâché et le bouton poussoir BP1 est appuyé.
- La diode D2 est allumée si le bouton poussoir BP0 et le bouton poussoir BP1 sont appuyés.
- La diode D3 est allumée si le bouton poussoir BP0 et le bouton poussoir BP1 sont relâchés.

Activité 1: Comportement combinatoire (Révisions - 10 min)

À partir du cahier des charges ci-dessus :

Question 1 Donner l'expression logique de chacune des diodes.

Étape 1 Implémenter le cahier des charges en utilisant les fonctions développées dans le TP 1.

2 La conversion Analogique Numérique

Ce TP a pour objectif de mettre en œuvre un CAN (Convertisseur Analogique Numérique) pour obtenir une valeur numérique image d'une grandeur analogique. Les fonctions que nous développerons dans ce TP pourront être utilisées dans les futurs TPs ou dans vos projets. A ce titre, il est important pour chacune d'entre elles de respecter les bonnes pratiques citées ci-dessous.

Bonnes pratiques à respecter pour chaque fonction

Pour chaque fonction développée, vous devrez documenter chacune de vos fonctions en utilisant la syntaxe vue au TP 1.

- ☐ respecter les conventions de nommage,
- ☐ avoir un nom explicite
- ☐ documenter la fonction :
  - ☐ en précisant son auteur et la date de création,
  - ☐ en précisant son rôle,
  - ☐ en précisant les paramètres d'entrée et de sortie,
- ☐ tester la fonction :
  - ☐ pour un fonctionnement normal
  - ☐ pour des cas limites

Activité 2: Création d'une bibliothèque (Préparation - 3 min)

Nous allons, tout d'abord, créer un dossier dans notre bibliothèque personnelle qui contiendra les fonctions liées au CAN.

Étape 2 Créer un dossier nommé `TP2_CAN` dans le répertoire `lib/`.

Étape 3 Créer deux fichiers `can.h` et `can.cpp` dans le dossier `lib/TP2_CAN`.

Étape 4 Dans le fichier `can.h`, ajouter la ou les ligne(s) permettant d'éviter les inclusions multiples.

Activité 3: Initialisation du CAN (10 min)

Question 2 Quelles sont les étapes nécessaires pour initialiser le CAN ?

Étape 5 Définir une fonction permettant l'initialisation du CAN.

Rappel : le prototype devra se trouver dans le fichier `can.h` et le corps de la fonction dans le fichier `can.cpp`.

Question 3 Dans quelle partie du programme (*setup()* ou *loop()*) devra-t-on appeler cette fonction ?

Cette fonction ne faisant rien d'autre qu'une initialisation, elle ne pourra pas être testée seule. Nous la testerons donc lors de l'implémentation des autres fonctions.

Activité 4: Lecture du CAN sur 8 bits (15 min)

Question 4 Quelles sont les étapes nécessaires pour lire une valeur analogique ?

Étape 6 Définir une fonction bloquante permettant la lecture d'une valeur analogique sur 8 bits.

Question 5 Compléter la checklist suivante pour cette fonction :

- ☐ Valeur pour une tension nulle : .....
- ☐ Valeur pour une tension maximale : .....
- ☐ Valeur à mi-course : .....

Activité 5: Test de la lecture du CAN sur 8 bits (5 min)

Étape 7 Écrire un programme permettant de tester la fonction de lecture du CAN en envoyant la valeur lue sur le port série.

Question 6 Vérifier les points de la checklist suivante puis faire vérifier par l'enseignant. Si nécessaire, corriger la fonction.

- ☐ Les fonctions d'initialisation et de conversion respectent les bonnes pratiques données en préambule.
- ☐ La fonction de lecture donne des valeurs cohérentes avec la position du potentiomètre (cf Activité précédente)
- ☐ Sans la fonction d'initialisation, la fonction de lecture ne fonctionne pas.

Étape 8 Archiver le programme dans votre dossier `TP2_CAN`.

Activité 6: Mesure du temps de conversion (5 min)

Question 7 D'après la documentation, quelle est la durée de conversion d'une valeur analogique ?

Étape 9 Dans votre programme, ajouter des lignes permettant de mesurer le temps de conversion de votre fonction et de l'afficher sur le port série.

Question 8 Quelle est la durée de conversion mesurée ? Est-elle en accord avec la valeur attendue ?

Activité 7: Erreur si temps de conversion trop long (10 min)

Étape 10 Modifier votre fonction de lecture pour qu'elle renvoie la valeur -1 si le temps de conversion est trop long.

Cahier des charges : led selon la valeur lue par le CAN

- Toutes les diodes sont éteintes si la tension est inférieure à 1 V.
- La diode D1 est allumée si la tension est comprise entre 1 V et 2 V.
- La diode D2 est allumée si la tension est comprise entre 2 V et 3 V.
- La diode D3 est allumée si la tension est comprise entre 3 V et 4 V.
- La diode D4 est allumée si la tension est supérieure à 4 V.

Activité 8: LED selon la valeur lue par le CAN (7 min)

Étape 11 Implémenter le cahier des charges ci-dessus en utilisant les fonctions développées.

Activité 9: Lecture du CAN sur 10 bits (15 min)

Question 9 Quelles sont les étapes supplémentaires nécessaires pour lire le résultat sur 10 bits ?

Étape 12 Définir une fonction bloquante permettant la lecture d'une valeur analogique sur 10 bits.

Question 10 Compléter la checklist suivante pour cette fonction :

- ☐ Valeur pour une tension nulle : .....
- ☐ Valeur pour une tension maximale : .....
- ☐ Valeur à mi-course : .....

Question 11 Vérifier les points de la checklist suivante puis faire vérifier par l'enseignant. Si nécessaire, corriger la fonction.

- ☐ La fonction respecte les bonnes pratiques données en préambule.
- ☐ La fonction de lecture donne des valeurs cohérentes avec la position du potentiomètre

Étape 13 Archiver le programme dans votre dossier `TP2_CAN`.

Activité 10: Mesure du temps de conversion sur 10 bits (5 min)

Étape 14 Mesurer le temps de conversion et le comparer avec la conversion 8 bits

3 Application : Utilisation du Joystick

Le Joystick de la carte est possède une résistance différente pour chaque axe. Il est donc possible de mesurer la position du joystick en utilisant le CAN.

Cahier des chrges : Utilisation du Joystick

- La diode D1 est allumée si le joystick est en position basse.
- La diode D2 est allumée si le joystick est en position haute.
- La diode D3 est allumée si le joystick est en position gauche.
- La diode D4 est allumée si le joystick est en position droite.

Activité 11: Valeurs du Joystick (15 min)

Étape 15 Implémenter le cahier des charges ci-dessus.

4 Création d'une classe

Activité 12: Création d'une classe (20 min)

Question 12 Qu'est-ce qu'une classe ?

Question 13 Quels sont les avantages de l'utilisation des classes ?

Étape 16 Créer une classe nommée `CConvAN10bits` dans le dossier `lib/TP2_CAN`.

Étape 17 Définir la fonction d'initialisation du CAN dans cette classe.

Étape 18 Définir la fonction de lecture sur 10 bits du CAN dans cette classe.

Étape 19 Définir le constructeur de cette classe. Il appellera la fonction d'initialisation du CAN.

Étape 20 Tester la classe en utilisant un programme similaire à celui de l'activité précédente.

Activité 13: Affichage de la direction du Joystick (15 min)

Le Joystick peut prendre les 8 positions suivantes :

- Bas
- Bas-Gauche
- Gauche
- Haut-Gauche
- Haut
- Haut-Droite
- Droite
- Bas-Droite

Étape 21 Afficher la direction du Joystick sur le port série en utilisant la classe définie.