

Terminale STI2D – Spé Système d'Information et Numérique (SIN)

Séance 4 – 18/09/2023

Rendu attendu

Ce TP devra faire l'objet d'un rendu qui ne sera pas noté. Le but pour moi est de visualiser comment vous rédigez vos comptes-rendus afin d'identifier les points d'amélioration pour les TP notés qui suivront.

Ce TP est long afin de permettre aux élèves les plus rapides d'avoir des challenges. Aussi, les derniers exercices font apparaître des composants que l'on n'a pas encore (re)vu ensemble.

Pour chaque exercice, il sera demandé :

- 1) le schéma électrique de l'exercice complet
- 2) le code de l'exercice complet.

Pour le code, vous créerez un répertoire nommé « SIN_TP_Arduino_NOM_Prenom » et vous y placerez à l'intérieur les fichiers de codes en « .ino » suivant la nomenclature « xx-nom-de-l-exercice » :

- xx : correspond au numéro de l'exercice en décimal, sur deux digit
- nom-de-l-exercice : le titre indiqué pour chaque exercice.

Chaque programme devra commencer par un en-tête en commentaire multi-lignes qui indiquera :

- nom et prénom
- date
- titre de l'exercice

Conseils et consignes générales

Pour chaque exercice, prenez le temps de lire l'ensemble des questions pour comprendre l'objectif final.

Pour chaque question, prenez le temps au brouillon de faire un algorithme ou un logigramme et le schéma électrique. **Il est important de conserver des traces de votre raisonnement pour y revenir en cas de problème lors de l'implémentation de votre solution.**

Si vous me demandez de l'aide car vous bloquez sur un point mais que vous n'êtes pas en mesure de me montrer votre raisonnement, je ne prendrai pas le temps de vous aider.

Vous disposez du matériel habituel : LEDs, boutons poussoir, fils électriques, résistances, etc. Pour ces dernières, vérifiez bien leur valeur grâce au code couleur des résistances. Vous trouverez ce code sur Internet.

Pensez à utiliser les résistances de pull-up pour absorber le bruit des boutons poussoir. Et la notion de PWM pour gérer les valeurs analogiques (luminosité progressive).

Il n'est pas nécessaire de me faire valider chaque question, mais **appelez-moi à chaque fin d'exercice**. Je passerai aussi vous voir individuellement pour suivre votre avancement.

Prenez garde à bien gérer les planchers et plafonds dans vos algorithmes (limites minimales ou maximales).

Pensez à commenter votre code.

Exercice 1 : chenillard

Dans cet exercice, vous programmerez un chenillard de 4 LEDs à trois commandes (trois BP).

Rappel sur le chenillard : chaque LED s'allume et s'éteint les unes après les autres. La durée où la LED est allumée doit être la même que la durée où toutes les LEDs sont éteintes.

- a) Premièrement, recréer un chenillard classique, sans contrainte.
- b) Le premier BP permettra de contrôler le sens du chenillard : un appui inverse le sens de défilement.
- c) Le deuxième BP permettra de réduire la vitesse du chenillard. La vitesse minimale (temporisation) doit être de 2 secondes. Le pas sera de 150ms.
- d) Le troisième BP permettra d'accélérer la vitesse du chenillard. La vitesse maximale (temporisation) doit être de 50 ms. Le pas sera de 150ms.

Exercice 2 : compteur-binaire

Dans cet exercice, vous programmerez un compteur binaire sur 4 bits (donc de 0 à 255). Vous utiliserez trois BP.

Chaque LED correspond à un bit. Une LED allumée signifie que le bit est à 1. Une LED éteinte signifie que le bit est à 0.

Exemple :

Objectif : coder la valeur binaire 0110

Organisation des LED :

LED 1

LED 2

LED 3

LED 4

LED1 = éteinte

LED2 = allumée

LED3 = allumée

LED4 = éteinte

- a) Créer l'algorithme qui permet de traduire un nombre décimal en binaire.
- b) Programmer l'allumage des leds en fonction d'un nombre binaire (faire une fonction indépendante)
- c) Le premier BP permettra une remise à zéro du compteur. Si le compteur est déjà à zéro (et seulement dans ce cas), un appui sur le BP mettra le compteur à son maximum, soit 255.
- d) Le deuxième BP permettra d'incrémenter le compteur.
- e) Le troisième BP permettra de décrémenter le compteur.

Exercice 3 : guirlande-noel

Dans cet exercice, vous programmerez une guirlande de Noël avec plusieurs modes. Chaque mode sera activé par un bouton-poussoir. Si le matériel le permet, créez un circuit à 6 LEDs minimum.

Pour chaque mode, un bouton-poussoir permettra de configurer la vitesse : soit rapide, soit lente.

- a) Mode 1 : clignotement alterné une LED sur deux.
- b) Mode 2 : clignotement alterné (une LED sur deux) mais avec une illumination progressive des LEDs (ainsi qu'une extinctions progressive).
- c) Mode 3 : clignotement alterné selon les couleurs des LEDs : d'abord les rouges, puis les jaunes, puis les vertes, puis les jaunes et on recommence.

Exercice 4 : potentiometre-0 (barregraphe)

Pour découvrir l'utilisation du potentiomètre, utilisez-le avec quatre LEDs. Les LEDs permettront de coder la position du potentiomètre (on utilise la variable p dans les règles ci-dessous) :

- 0 LED allumée : $0 \leq p < 20 \%$
- 1 LED allumée : $20 \leq p < 40 \%$
- 2 LEDs allumées : $40 \leq p < 60 \%$
- 3 LEDs allumées : $60 \leq p < 80 \%$
- 4 LEDs allumées : $80 \leq p < 100 \%$

Exercice 5 : potentiometre-1

Reprenez l'exercice 1, mais au lieu d'utiliser deux BP pour gérer la vitesse du chenillard, vous utilisez un potentiomètre.

Exercice 6 : potentiometre-2

Reprenez l'exercice 2, le potentiomètre sera l'entrée qui définira la valeur du compteur, toujours de 0 à 255.

Exercice 7 : afficheur-7-segments

Pour découvrir l'utilisation de l'afficheur 7 segments, réalisez un premier programme qui affichera chaque segment les uns après les autres dans l'ordre suivant : A, B, C, D, E, F, G, DP.

Exercice 8 : compteur-decimal

Créer un compteur décimal en se basant sur l'exercice 2. La sortie sera sur un afficheur 7 segments. Les entrées seront contrôlées par des boutons poussoir : remise à zéro, incrémentation, décrémentation.