

IV - UTILISATION

Ce document se trouve sur ce site : http://www.la-tour.info/uts/uts_page15.html#conduite

Si vous réaliser des améliorations matériels ou logicielles, merci de les poster sur le forum RMF pour en faire profiter le plus grand nombre.

Si vous publier cette réalisation avec ou sans améliorations, vous devez publier votre code source.

Ce logiciel est un logiciel libre. Exigence du concepteur : Ne pas modifier la ligne d'affichage "JLF xx/xx/xxxx" sur l'écran LCD.

On peut modifier ce programme et le diffuser. Dans ce cas, il faut préciser l'origine et donner accès aux sources modifiées.

Définition : Un logiciel libre est un logiciel distribué avec l'intégralité de ses programmes-sources, afin que l'ensemble des utilisateurs qui l'emploient, puissent l'enrichir et le redistribuer à leur tour.

Note : Un logiciel libre n'est pas nécessairement gratuit et les droits de la chaîne des auteurs sont préservés.

Équivalent étranger : free software, open source software.

(Source : Vocabulaire de l'informatique (liste de termes, expressions et définitions adoptés), NOR: CTNX0710138K, J.O n° 93 du 20 avril 2007 page 7078, texte n° 84)

logiciel libre

Par logiciel libre on entend un logiciel qui offre la liberté aux utilisateurs d'exécuter, de copier, de distribuer, d'étudier, de modifier et d'améliorer le logiciel. Plus précisément, elle fait référence à quatre types de liberté pour les utilisateurs du logiciel :

La liberté d'exécuter le programme, pour tous les usages (liberté 0).

La liberté d'étudier comment le programme fonctionne et de l'adapter à ses besoins (liberté 1). L'accès au code source est une condition requise.

La liberté de redistribuer des copies, (liberté 2).

La liberté d'améliorer le programme et de diffuser les améliorations au public pour en faire profiter toute la communauté (liberté 3). L'accès au code source est une condition requise.

Mise en route

La carte Arduino a été testée et les fichiers de configuration de l'interface installés.

1 / Brancher les deux ports usb de l'Arduino, à l'ordinateur.

Cliquer sur : Menu Windows > Paramètres > Périphériques.

On trouve la ligne : Autres appareils > Arduino DUE Programming Port (COMXX).

Ce numéro de port COMXX, doit être celui renseigné dans le fichier : TSCClassic Raildriver and Joystick Interface/KeysMaps/Ports/ AssignedPorts.txt.

Le port "Native usb Sam3x" assure l'émulation d'un second clavier, pour envoyer des codes de touche au simulateur, comme le fait le clavier principal.

2 / On démarre le programme "TSCClassic Raildriver and Joystick Interface V3.3.0.7", avec les fichiers de configuration corrects.

On clique sur le bouton [Start Call back].

On doit avoir dans la fenêtre centrale "Data received from Railworks", toutes les variables attendues par la carte Arduino.

1 / Menu de test des positions des servomoteurs

(1) SERVOMOTEURS

TEST POSITIONS

Ce menu permet d'étalonner les équipements utilisant les servomoteurs.

On établit les caractéristiques d'un manomètre modifié avec un servomoteur.

⇒ Attention, les servomoteurs standards acceptent des impulsions entre 1000 et 2000 µsec. Utiliser des impulsions en dehors de ces limites peut endommager les servomoteurs. Vérifier que les moteurs ne se trouvent pas en butée.

Mes servomoteurs, équipés de PTK-7350MG-D, acceptent une impulsion de 700 à 2300 µsec.

J'envoie une impulsion de 700 µs. Je retire l'aiguille pour la repositionner à 0 Bar.

Mon manomètre affichera donc 0 Bar, quand il recevra une impulsion de 700 µs.

Ensuite, on envoie une impulsion de 2300 µsec.

Je note la nouvelle valeur affichée, par exemple 10 Bars.

Cela me permettra de calculer les formules dans le programme de réception des données des manomètres.

Exemple de calcul pour "RE_control" :

```
if (!strcmp(buffer_rx[1], "RE_control")) {  
    buffer_rx_fvaleur = atof(buffer_rx[2]); // Conversion du dernier champ : Ascii-> Flottant.  
    buffer_rx_fvaleur = 160*buffer_rx_fvaleur; // 5.5 Bars maximum reçu : Cadran marqué 0 Bar = 700 µs à 10 Bars pour 2300 µs : Coefficient = (2300-  
700)/(9,5) = 160  
    buffer_rx_valeur = 700 + int(buffer_rx_fvaleur); // Valeur minimum = 700 µsec. On passe en valeur entière pour la suite des calculs.
```

⇒ Il faudra aussi renseigner les valeurs "i2c_srv_init[]" dans le programme principal :

```
const int i2c_srv_init[] = {700, 700, 700, 700, 700, 700, 700, 700}; // Pour les 8 sorties SERVO du n° 0 au n° 7.
```

C'est pour positionner les servomoteurs à 0 au démarrage de l'Arduino.

Renseigner obligatoirement 8 valeurs en µs pour les 8 servomoteurs. Valeur en µsec = 600 µs minimum à 2400 µs maximum. Par défaut mettre 700 µs ou 2200 µs.

Menu de test des positions des servomoteurs

SRV: x VAL: yyyy

INCREMENT : zzz

SRV : x : Servomoteur en cours = n° x.

VAL: yyyy : Valeur en µsec envoyée au servomoteur n° x. De 500 µs à 2500 µs.

INCREMENT : zzz : Valeur du facteur d'incrément en cours.

Boutons :

- (RAZ) = Retour au menu d'accueil.
- (Précédent) = Passe au servomoteur précédent.
- (Suivant) = Passe au servomoteur suivant.
- (Facteur d'incrément) = Modifie le facteur d'incrément de 1, 10 à 100.

Bouton rotatif : Modifie la valeur envoyée, en fonction du facteur d'incrément.

Si l'on met 100, la valeur envoyée est modifiée de +/- 100 par ce bouton rotatif.

2 / Menu de test des positions des sorties pwm

(2) GALVANOMETRES

TEST POSITIONS

Ce menu permet d'étalonner les équipements utilisant les sorties pwm.

On peut étalonner un galvanomètre modifié pour le 5 Volts.

⇒ Il faut avoir au préalable renseigné les valeurs "i2c_pwm_inv[]" dans le programme principal :

```
const char i2c_pwm_inv[] = {'I', 'N', 'N', 'N', 'I', 'I', 'N', 'N'}; // Pour configurer les 8 sorties PWM, du n° 0 au n° 7.
```

Il faut déclarer obligatoirement les 8 sorties PWM en 'N'ormale ou 'I'nverse.

C'est pour bien positionner la sortie de ces équipements au démarrage de l'Arduino, et les animer dans le bon sens.

Si impulsion est 'N'ormale : Galva amplifié, Carte Triphasé pour Compte-tours ou Tachro, renseigner 'N'.

Si impulsion est 'I'nversée : Galva non amplifié relié au (+), renseigner 'I'.

Renseigner obligatoirement ces 8 valeurs.

Sortie de type 'N' : Quand on envoie '0' => 0 Volt. Quand on envoie '4095' => 5 Volts.

Sortie de type 'I' : Quand on envoie '0' => 5 Volts. Quand on envoie '4095' => 0 Volt.

Ca permet d'étalonner un galvanomètre, en envoyant 0, l'aiguille doit être à sa position d'origine.

Si l'on ne peut pas corriger un écart, on note la valeur à ajouter dans les calculs pour correction.

En envoyant 4095, soit le maximum possible, on peut étalonner les galvanomètres équipés d'un réglage.

Pour avoir le meilleur fonctionnement possible, il faut régler les galvanomètres au maximum de leur échelle utile.

Il faudra noter les valeurs maximum envoyées par l'ordinateur. Par exemple, un maximum de 1,5 KV pour un appareil allant jusqu'à 2,0 KV. L'échelle utile est 1,5 KV.

On envoie alors 4095 sur cette sortie, et pour un maximum affichable de 1,5 KV, on règle le galvanomètre sur 1,5 KV, même si il peut monter à 2,0 KV. Comme cela la résolution du montage est optimum.

Menu de test des positions des sorties pwm

PWM: x VAL: yyyy

INCREMENT : zzzz

SRV : x : Servomoteur en cours = n° x.

VAL: yyyy : Valeur entre 0 et 4095 envoyé en sortie pwm.

INCREMENT : zzz : Valeur du facteur d'incrément en cours.

Boutons :

- (RAZ) = Retour au menu d'accueil.
- (Précédent) = Passe à la sortie pwm précédente.
- (Suivant) = Passe à la sortie pwm suivante.
- (Facteur d'incrément) = Modifie le facteur d'incrément de 1, 10, 100 à 1000.

Bouton rotatif : Modifie la valeur envoyée, en fonction du facteur d'incrément.

Si l'on met 100, la valeur envoyée est modifiée de +/- 100 par ce bouton rotatif.

⇒ La tension en sortie dépend de la valeurs "i2c_pwm_inv[]" renseignée pour cette sortie, dans le programme principal :

3 / Menu d'affichage des erreurs

(3) AFFICHE LES ERREURS

Si le programme détecte une erreur à l'initialisation, il allume la led rouge "Erreur".
Dans ce menu, on affiche les erreurs en cours.

Les erreurs possibles sont :

MODULE PCA9685 MODULE MCP23017
SERVO/PWM ABSENT ENTREES1: ABSENT

Indique en clair l'erreur et le numéro du module.
Vérifier les alimentations
Vérifier le câblage et la bonne insertion des plots et barrettes.
Vérifier les adresses I2C réalisée par soudure.

Boutons :

- (RAZ) = Retour au menu d'accueil.

Bouton rotatif : Affiche les autres erreurs.

4 / Menu d'affichage des informations

(4) INFORMATIONS SYSTEME

Affiche la durée d'exécution du programme :

DUREE BOUCLE :
30 μ s

Affiche la durée moyenne du passage complet dans le programme en μ s, sur une moyenne de 1000 passages.
On doit avoir une durée de 40 μ s maximum.

Boutons :

- (RAZ) = Retour au menu d'accueil.

Bouton rotatif : Affiche le sous menu suivant

Affiche la valeur analogique du boitier de configuration :

TOUCHES DU LCD
938<x<1023: 1023

Les boutons du boitier de configuration sont reliés à une entrée analogique. Cela permet d'avoir un seul fil, pour lire tous ces boutons.

On teste le bon fonctionnement des boutons.

Tourner le bouton rotatif pour tester les boutons.

En appuyant sur un de ces boutons, on a :

TOUCHES DU LCD 1 N° du bouton appuyé de 1 à 5.
597<x<767: 717 Valeur analogique lue.

Sans appuyer sur un bouton, on doit avoir environ 1023.

En appuyant sur un bouton, on a une valeur qui s'affiche, ici 717, qui doit être comprise entre 597 et 767.

Le n° du bouton retenu s'affiche, ici '1'.

Pour quitter ce menu, tourner le bouton, pour repasser au menu précédent.

Bouton rotatif : Affiche le sous menu précédent ou suivant

Affiche le nombre de code de touche qu'il reste à envoyer :

TCH BUFFER : 2



Affiche l'état de la file d'attente d'envoi de touche du clavier. Le bargraphe fonctionne entre 0 et 16 touches. Le buffer peut contenir 70 touches.

Boutons :

- (RAZ) = Retour au menu d'accueil.

Bouton rotatif : Affiche le sous menu précédent

5 / Menu d'affichage des valeurs

(5) AFFICHE VAL.

RECUES/ENVOYEES

Tourner le bouton rotatif, pour afficher les servomoteurs, les sorties pwm, les sorties tout ou rien et les entrées.

On affiche les valeurs des servomoteurs :

SERVO INIT : 700 Valeur en μ sec, à l'initialisation du programme.

SRV: 0 (μ s) : 1500 N° du servomoteur et valeur en cours en μ sec.

Boutons :

- (RAZ) = Retour au menu d'accueil.

Bouton rotatif : Affiche le servomoteur précédent ou suivant

On affiche les valeurs des sorties pwm :

PWM/GALVA/CT : I Sortie de type : 'N' = Normale, ou 'I' = Inversée.

PWM: 0 VAL: 2048 N° de la sortie pwm et valeur en cours entre 0 et 4095.

Boutons :

- (RAZ) = Retour au menu d'accueil.

Bouton rotatif : Affiche la sortie pwm précédente ou suivante

On affiche les valeurs des sorties tout ou rien :

SORTIES*16 : x Bloc de sortie n° x.

0000000000000001 Sorties à 0 ou 1. Ici la sortie n° 0 est active.

Boutons :

- (RAZ) = Retour au menu d'accueil.

Bouton rotatif : Affiche la sortie précédente ou suivante

On affiche les valeurs des entrées :

16*ENTREES : x Bloc d'entrée n° x.

0000000000000001 Entrées à 0 ou 1. Ici l'entrée n° 0 est active.

Boutons :

- (RAZ) = Retour au menu d'accueil.

Bouton rotatif : Affiche la sortie précédente ou suivante

On affiche les caractères envoyés à l'ordinateur :

TCH ENV CODE : 98 Affiche le n° ascii en décimal. Exemple 'b' = 98.

NORMAL : 400 : b Type de caractère : Type d'envoi : Caractère ascii.

Type de caractère : NORMAL, CTRL, SHIFT ou CTRL+SHI(FT).

Forme d'envoi : PRESS = Uniquement début de touche, RELE(ASE) = Uniquement fin de touche ou XXXX = Envoi du début et de fin de touche espacé de XXXX msec.

Caractère ascii si affichable. ^f = Ctrl+'f'.

Boutons :

- (RAZ) = Retour au menu d'accueil.

Bouton rotatif : Affiche la sortie précédente ou suivante

On affiche les caractères échangés avec le KVB :

KVB:PAS DE RECPT Pas de caractère reçu ou envoyé.

KVB:PAS D'ENVOI

VERS -> KVB: x Affiche le dernier caractère ascii envoyé vers le KVB.

DEPUIS <- KVB: y Affiche le dernier caractère ascii reçu depuis le KVB.

⇒ **ATTENTION : Pour un fonctionnement nominal, l'écran LCD doit afficher le menu d'accueil "JLF – 12/05/2025 MENU:TOURNER BTN"**

Si un autre texte est affiché, il y a une forte baisse des performances, car animer l'écran LCD prend beaucoup de temps.

Le pupitre force l'état du simulateur.

Si les interrupteurs du pupitre sont correctement configurés, ils font bouger les interrupteurs sur l'ordinateur.

Si les retours d'information sont bien configurés, si l'on bouge un interrupteur sur l'ordinateur, il doit revenir tout seul à sa bonne position, en moins de 3 secondes.

C'est les interrupteurs du pupitre physique qui forcent la position des interrupteurs sur l'ordinateur.