

# Bras robotisé et tri de blocs selon le poids

PROJET DE FIN DE SESSION

François Ruelland,  
Gabriel Fortin-Bélanger et Kevin Cotton

Session 3

Novembre 2025

## Table des matières

1.	Description générale .....	3
2.	Descriptions des tâches à faire pour le PIC .....	5
3.	Descriptions des tâches à faire par le STM32 .....	5
4.	Descriptions des tâches à faire pour la balance .....	5
5.	Descriptions des tâches à faire pour le PCB du Contrôleur PWM .....	6
6.	Principe de fonctionnement des programmes.....	6
7.	Algorithme du PIC.....	7
8.	Programme du STM32 .....	8
9.	Remise de l'ensemble du projet .....	11
10.	Horaire du projet.....	12
11.	Équipes .....	12
12.	Évaluation .....	13
13.	Rapport (50%).....	14
14.	Préparation à L'ASP.....	15
ANNEXE 1	Schéma de la carte Driver PWM.....	16

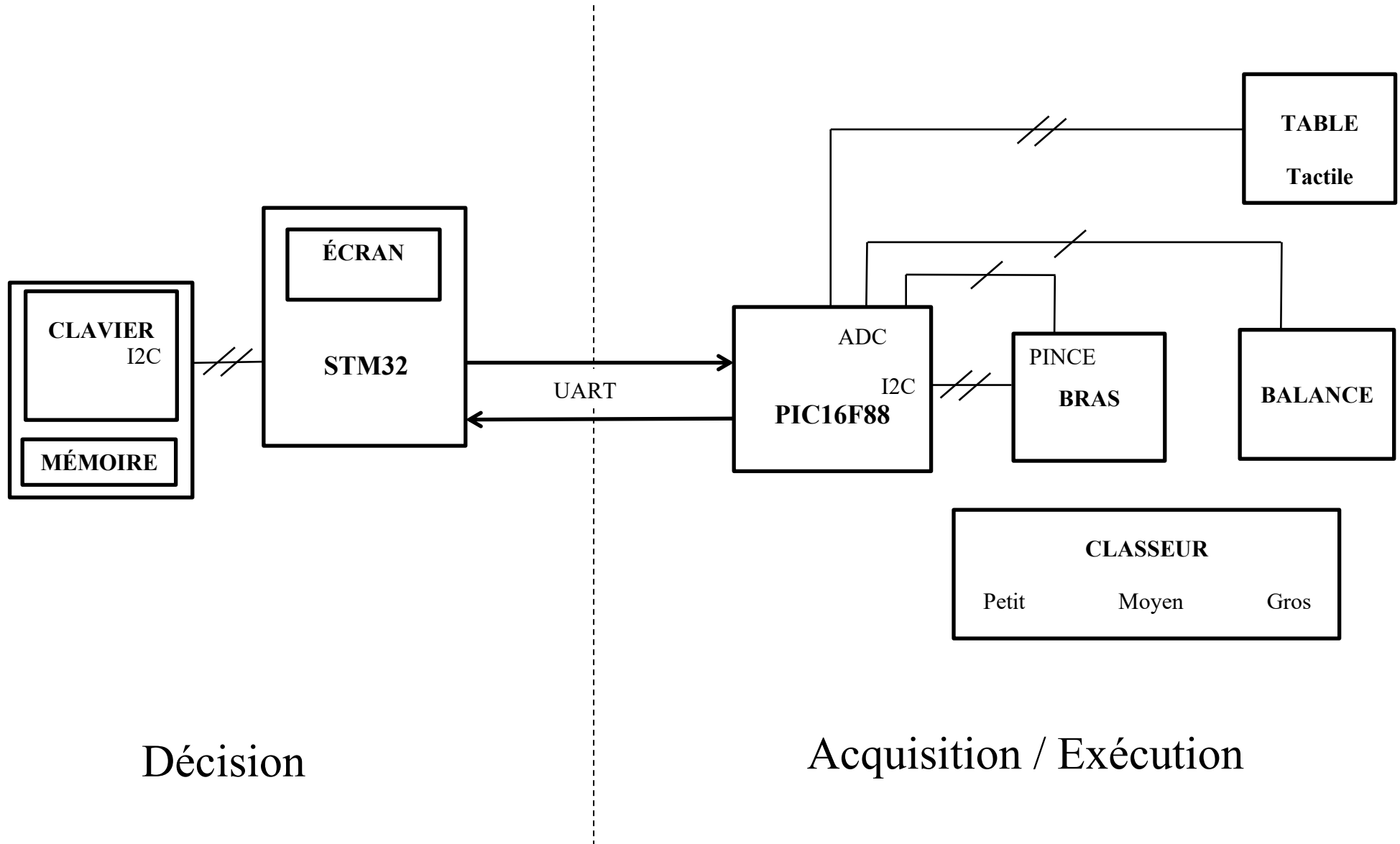
# 1. Description générale

Le projet consiste à utiliser un bras robotisé pour permettre le tri de blocs. Les blocs utilisés seront de poids différents. Pour permettre de reconnaître les différents blocs, une balance sera employée.

Le fonctionnement du projet sera le suivant :

- Un bloc sera déposé sur une membrane tactile.
- Le système détectera la position du bloc et placera le bras pour saisir le bloc.
- Un capteur sur la pince permettra de savoir si effectivement le bloc a été saisi.
- Le bloc sera ensuite déposé sur la balance pour déterminer son poids.
- Finalement, selon le poids du bloc, le bras ira la déposer au bon endroit dans le classeur.
- Trois positions sur la membrane tactile seront assignées pour chaque équipe.

## Vue d'ensemble du projet



## 2. Descriptions des tâches à faire pour le PIC

Voici la liste des tâches à faire dans le programme de la carte de développement PIC.

- Lire la position active de la membrane tactile.
- Lire la pression de la pince.
- Lire le poids sur la balance.
- Transmettre à tous les 100 ms au STM32:
  - La valeur de la position active de la membrane tactile (X et Y).
  - La pression exercée par la pince.
  - Le poids présent sur la balance.
- Recevoir du STM32 les positions des moteurs. Ces positions sont envoyées par le STM32 aussitôt qu'il a reçu les données du PIC.
- Transmettre au contrôleur de PWM, en I2C, les nouvelles positions des moteurs.

Le PIC ne prend aucune décision. Il lit l'information du système, ensuite transfère le tout au STM32 et celui-ci retourne les nouvelles positions désirées pour les moteurs. Le PIC transfère alors ces positions au contrôleur PWM.

## 3. Descriptions des tâches à faire par le STM32

Voici la liste des tâches à faire dans le programme de la carte STM32.

- Lire le clavier I<sup>2</sup>C pour recevoir des commandes (déplacement des moteurs du bras).
- Recevoir du PIC :
  - Les informations sur la position active de la membrane tactile (X, Y).
  - Les informations sur la pression exercée par la pince.
- Afficher sur l'écran LCD l'information du système (X, Y, Pince, Balance).
- Afficher les valeurs voulues des positions des moteurs.
- Obtenir le poids des blocs sur la balance (ADC)
- Transmettre au PIC, par RS-232, les nouvelles positions des moteurs, après avoir reçu les informations venant du PIC.

C'est le STM32 qui prend l'ensemble des décisions pour le déplacement du bras.

## 4. Descriptions des tâches à faire pour la balance

Pour lire la balance, vous devrez monter sur le proto présent sur la plaque du bras, un circuit permettant de conditionner le signal venant d'une jauge de contrainte. Pour y arriver, vous devrez faire les étapes suivantes (voir document sur la jauge de contrainte) :

- Mesurer le signal (tension) que la jauge vous donne avec les trois types de blocs.
- Établir le gain nécessaire pour obtenir des tensions mesurables avec les convertisseurs du STM32.
- Monter sur le proto votre circuit.
- Relier la sortie de votre circuit à une entrée analogique du STM32.

## 5. Descriptions des tâches à faire pour le PCB du Contrôleur PWM

Afin de relier le PIC au servomoteur, nous utiliserons le module Contrôleur de Servomoteur PCA9685. Pour permettre l'interfaçage entre le PIC et le module, deux 'PCB-Connecteurs' sont fournis et installés.

- Schéma fourni (Annexe 1), PCB qui est branché sur le module Contrôleur de Servomoteur.

## 6. Principe de fonctionnement des programmes

Il y aura une communication UART entre le STM32 et le PIC.

Afin de limiter le plus possible les problèmes, nous synchroniserons les échanges pour nous assurer que les deux systèmes ne transmettent pas en même temps.

Le maître des échanges sera le PIC. Il transmettra une trame au STM32 à toutes les 100ms ( $\pm 10$ ms). Lorsque le STM32 reçoit une trame valide, il retourne une trame au PIC. La longueur des trames sera limitée à 8 octets.

Donc, le PIC transmet une "photo" du système (X, Y, Pression Pince et Poids sur la Balance) à toutes les 100ms. Après avoir reçu cette trame, le STM32 transmettra la position désirée pour les cinq moteurs.

Au départ le STM32 transmettra la position médiane pour tous les moteurs sauf Pince (ouverte). Le bras devra donc se positionner à la verticale avec la pince ouverte.

Deux types d'événements pourront faire déplacer le Bras.

- Premièrement, à l'aide du clavier I2C, il sera possible de faire bouger un moteur à la fois.
- Deuxièmement, si une des 3 cases qui vous est attribuée sur la membrane tactile est pressée. À ce moment on entre en mode de déplacement automatique. Il faudra alors exécuter trois séquences.
  - Recherche du bloc sur la case pressée.
  - Allez sur la balance pour peser le bloc afin de déterminer le poids.
  - Dépôt du bloc sur le classeur au bon endroit selon le poids et retour du bras dans la position de repos (Pince près de la plaque au pied de la base).

Lors de l'exécution d'une séquence, il faudra prévoir un délai après chaque petit mouvement du bras pour permettre aux moteurs d'atteindre leur position (rappelez-vous POB).

En tout, sept séquences seront mémorisées. Trois positions différentes sur la membrane tactile (vos cases), une séquence pour aller à la balance et peser le bloc et finalement, trois séquences pour se rendre sur une des trois cases du dépôt. Chaque séquence pourra avoir 10 mouvements pour y arriver.

## 7. Algorithme du PIC

### Algorithme sommaire du programme dans le PIC

```
DÉBUT
  InitPIC().
  I2CResetPCA9685().           ; Reset software.
  I2CSetFreqPCA9685().         ; Frequence a 50Hz.
  Boucle = 127 * 256.           ; 2 variables.
  TANT QUE (VRAI)
    Boucle--.
    SI(Boucle == 0)             ; ~100ms?
      Boucle = 127 * 256.
      LireMembraneTactile().    ; Lit X et Y.
      Pince = LirePince().      ; Lit la pression sur la Pince.
      Balance = LireBalance().  ; Lit le poids sur la balance.
      TransmetTrame();          ; Transmet X, Y, Pince, Balance.
    FIN SI. (Boucle == 0)
    SI(Caractere reçu sur le Port serie)
      Car = LirePortSerie().
      SI(Car == 'G')
        Car = LirePortSerie().
        SI(Car == 'O')
          Moteur0 = LirePortSerie(). ; Position du Moteur 0.
          Moteur1 = LirePortSerie().
          Moteur2 = LirePortSerie().
          Moteur3 = LirePortSerie().
          Moteur4 = LirePortSerie().
          CheckSum = LirePortSerie().
          SI(Trame Valide)         ; (Somme Octets 0 a 6) - CheckSum = 0
            I2CSetPWMPKA9685(0, Moteur0). ; Positionne les moteurs.
            I2CSetPWMPKA9685(1, Moteur1).
            I2CSetPWMPKA9685(2, Moteur2).
            I2CSetPWMPKA9685(3, Moteur3).
            I2CSetPWMPKA9685(4, Moteur4).
          FIN SI. (Trame Valide)
        FIN SI. (Car == 'O')
      FIN SI. (Car == 'G')
    FIN SI. (Caractere reçu)
  FIN TANT QUE.
FIN DU PROGRAMME.
```

Pour le fonctionnement des servomoteurs et de la membrane tactile, voir les documents remis.

Voici la structure de la trame transmise par le circuit PIC:

Octet 0	Octet 1	Octet 2	Octet 3	Octet 4	Octet 5	Octet 6	Octet 7
'G'	'O'	X	Y	Pince	Balance	RESERVE	Checksum

Octet 0 : 'G' (Start) Indique un début de trame. On attend un 'O' pour valider.

Octet 1 : 'O' (Start) pour valider le début d'une trame.

Octet 2 : Valeur en X de la membrane tactile.

Octet 3 : Valeur en Y de la membrane tactile.

Octet 4 : Valeur de la pression sur la Pince.

Octet 5 : Valeur de la Balance (poids).

Octet 6 : Réserve.

Octet 7 : CheckSum (Somme des 7 premiers octets (sur 8 bits), sans complément)

## 8. Programme du STM32

Votre projet sera formé de plusieurs fichiers sources (chaque fichier .C sera couplé à un fichier .H):

### Utilité des touches du clavier I<sup>2</sup>C en mode manuel

Touche	Fonction
1	Sélectionne le moteur 1 (Base) pour le contrôle avec les touches # et *
2	Sélectionne le moteur 2 (Épaule) pour le contrôle avec les touches # et *
3	Sélectionne le moteur 3 (Coude) pour le contrôle avec les touches # et *
4	Sélectionne le moteur 4 (Poignet) pour le contrôle avec les touches # et *
5	Sélectionne le moteur 4 (Pince) pour le contrôle avec les touches # et *
0	Change la valeur d'incrémentement des moteurs. Suggestion : Garder la valeur entre 1 et 16. Pourrait être incrémenté de plus de 1 à la fois.
*	Baisse la partie du bras choisie par les touches de 1 à 5 en utilisant la valeur d'incrémentement choisie par la touche 0. Dans le cas de moteur 1, on tourne à gauche. Les valeurs doivent rester entre 0 et 205.
#	Monte la partie du bras choisie par les touches de 1 à 5 en utilisant la valeur d'incrémentement choisie par la touche 0. Dans le cas de moteur 1, on tourne à droite. Les valeurs doivent rester entre 0 et 205.
A	Partir une serie de mouvements qui permettrons de réchauffer les moteurs.

D'autres touches pourront être définies afin de faire du déverminage comme Arrêter / Démarrer une séquence, avancer étape par étape dans une séquence, etc.

### Exécution du mode automatique

#### *Sauvegarde des positions.*

#### *Position de repos*

Au début, le bras se tient droit, en pointant dans les airs.

Si après la première séquence il n'y a pas de bloc détecté, il faut remettre le bras en position de repos.

Si après la pesée, aucun bloc n'est détecté, il faut remettre le bras en position de repos.

Après avoir posé le bloc dans le classeur, le bras retourne en position de repos.

### Réception des données venant du PIC

- La réception devra être traitée en interruption afin de ne rien manquer.
- C'est dans cette fonction que les 8 octets seront analysés et mis dans un buffer de réception et que le « Checksum » sera vérifié.
- Un drapeau sera levé si la réception est correcte afin d'initier la transmission vers le PIC.



## Structure de la trame transmise par le STM32

Octet 0	Octet 1	Octet 2	Octet 3	Octet 4	Octet 5	Octet 6	Octet 7
'G'	'O'	Moteur0	Moteur1	Moteur2	Moteur3	Moteur4	Checksum

Octet 0 et Octet 1 : 'G' et 'O' Indique un début de trame.

Octet 2 à Octet 6 : Position désirée (0 à 205) pour les Moteur (0 à 4).

Octet 7 : CheckSum (Somme des octets 0 à 6 (sur 8 bits) sans complément).

## Autres requis

### *Utilisation de structures*

Vous devez utiliser au minimum une structure qui contient la position des cinq moteurs. Vous pouvez bien sûr faire plusieurs structures au besoin.

### *Affichage :*

L'affichage ressemble à ceci :

```
1:66 2:66 3:66 4:66
5:66 X:FF Y:FF P:FF
B:FF Bloc:--
05 Seq:0 Step:0 OffL
```

- 1 à 5 représente la position de moteurs
- X et Y représente les valeurs de la table tactile
- P est pour la valeur analogique de la pince
- B représente la valeur analogique de la balance
- Bloc est le type de blocs : Pe (Petit), Mo (Moyen), Gr (gros) et -- (pas de bloc).
- Le 05 est la valeur de l'incrément
- seq est la séquence active (de 1 à 7)
- L'étape (déplacement) dans la séquence (de 1 à 10)
- L'état de la communication (OffL ou OnL) (pourrait aussi être un caractère spécial animé).

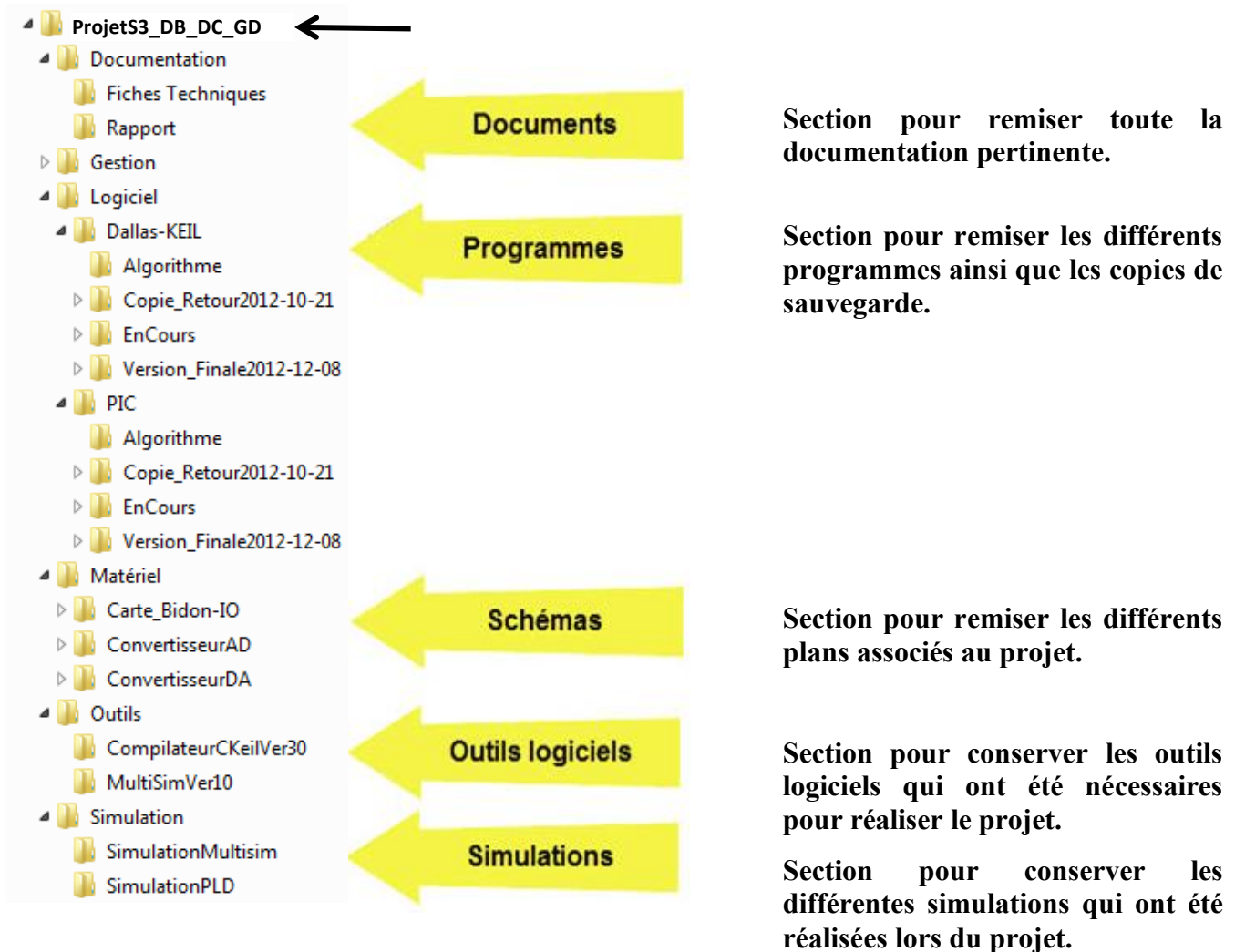
## 9. Remise de l'ensemble du projet

Vous devez nous remettre l'ensemble de votre projet dans la structure de répertoire qui correspond à la représentation de la figure qui suit.

Cet exemple illustre le cas d'un projet qui peut être plus complexe que le vôtre. Les répertoires de Gestion et de Simulation pourraient être enlevés si pas utilisé.

Exemple, dans le répertoire STM32-Keil. On retrouve des copies de sauvegarde (Copie\_Retour), une dernière version sur laquelle on travaille (EnCours) et lors de la remise finale une dernière copie (version\_Finale) nous indiquant celle qu'il faut regarder pour la correction. C'est le même principe qui est utilisé dans les autres répertoires.

Votre Dossier "ProjetS3\_VosInitiales", sera remis dans les dossiers suivants du disque Pedagogie:  
P:\243-Technologie du Genie Electrique\!Depot\TSO\ProjetS3.



### Liste des différents répertoires pour la réalisation d'un projet.

- Section pour les documents (Documentation).
- Section pour la gestion ou planification du projet (Gestion).
- Section pour les programmes (Logiciel).
- Section pour les schémas (Matériel).
- Section pour les outils logiciels de travail (Outils).
- Section pour les simulations (Simulation).

## 10. Horaire du projet

Événement	Local	Dates
Présentation du projet – Cours théorique du cours 247-336	Q2104	25 novembre 2025
Avancement du projet	Q2233	Périodes de laboratoire des 3 cours
Avancement du projet et quelques notions si nécessaire	Q2233	Périodes théoriques des 3 cours
Examen du cours 243-376		28 novembre 2025
Examen du cours 243-356		4 décembre 2025 12h
Examen du cours 243-336		9 décembre 2025
Présentation des projets par les étudiants	Q2233	12 décembre 2025
Date limite de remise du rapport		17 décembre 2024 à 23h59

## 11. Équipes

ÉQUIPE	Étudiant # 1	Étudiant # 2	Étudiant # 3	Étudiant # 4	Cases
1	Takoukam Touole, Rowan Ramid	Noiseux, Sédrick	Tafopih Keunang, Ilya Batherote		
2	Cazzolato, Thomas	Di Stefano, Jessica	Bolduc, Javier		
3	Bou Boinne, Arthur	Tsapi Takoubo, Stephane	Vera Boada, Laura Valentina		
4	Bouchard-Cloutier, Charlélie	Vézina, Maxime	Quito Arias, Edwin Yobany		
5	MacDonald, William	Perron, Antoine	Bégin, Hadéyss		
6	Desrosiers, David	Soulard, Mathieu	Charroin, Louison		
7	Gironcel, Jonathan Jules Roch	Caron, Samuel	Muntukiese, Jonathan (S)	Palanca, Florian	

## 12. Évaluation

COMPÉTENCES: E,F,G,H,J,M,P,Q,S,T,U	Fonctionnement		Total
Trame RS-232 envoyée par le STM32	/2		
Trame RS-232 reçu par le STM32	/2		
Gestion manuelle avec clavier (STM32)	/2		
Affichage LCD (X, Y, Pince, Balance)	/2		
Position initiale du bras	/2		
Récupération sur les cases assignées	/3		/
Dépôt du bloc sur la balance	/3		
Lecture du bloc sur la balance	/2		
Identification correcte des blocs	/2		
Dépôt du bloc dans la bonne case	/3		
Retour du bras à la position de repos	/2		
Présentation, démonstration, questions	/6		
Qualité du montage sur proto (balance)	/3		
COMPÉTENCES : C,D,L,N,S,V		<b>Rapport</b>	
Page titre		/1	
Table des matières		/2	
Préparation		/5	
Description du fonctionnement		/10	
Description de l'utilisation		/6	/
Code source .C		/7	
Conclusion du projet		/4	
Conclusion de SESSION (individuelle):		/3 /3 /3	
<b>Options</b>			
	/	/	
<b>TOTAL</b>	<b>Fonctionnement</b>	<b>Rapport</b>	
			/

## 13. Rapport (50%)

Voici ce que le rapport doit contenir :

- Page titre
- Table des matières
- Préparation :
  - Méthodes ou techniques utilisées pour arriver à vos fins
  - Principales fonctions et variables utilisées
- Description du fonctionnement
  - Schéma Bloc qui schématise l'architecture du système complet incluant les parties logicielles et matérielles.
- Description de l'utilisation
  - Procédure à suivre pour faire fonctionner le montage
- Programmes
  - Dans les répertoires
- Conclusion du projet
  - Ce que vous avez appris dans ce projet, ce que ça vous a apporté
  - Ce qui a mal fonctionné, vos solutions et hypothèses
- Conclusion de SESSION (individuelle)
  - Ce que vous avez apprécié et moins apprécié durant cette session
  - Ce qui a bien et moins bien été durant cette session

## 14. Préparation à L'ASP

Dans le but de préparer l'étudiant à l'ASP, il est convenu de rencontrer chacun d'eux, individuellement, à la fin des présentations de projets. Lors de cette rencontre, l'emploi des grilles (voir pages suivantes) permettra d'attester l'atteinte par l'étudiant des grands résultats visés dans la formation.

Cette étape précieuse de l'évaluation de l'étudiant lui permettra de se situer à un moment clé de sa formation, soit lors du projet de fin de session.

Grille d'observation				
SESSION #3				
NOM DE L'ÉTUDIANT : _____	Semaine : _____			
<b>PLANIFICATION (M)</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>
• Planifier l'élaboration de programmes structurés écrit en assembleur. (M)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
•	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>CONCEPTION (C, D, L, N, S et V)</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>
• Concevoir un circuit imprimé. (N, V)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
• Résoudre des problèmes de logique de programmation d'un système ordonné utilisant un microcontrôleur communiquant avec des circuits mis en réseau multipoint à protocole simple. (S)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
• Concevoir des interfaces numériques. (V)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
• Concevoir des interfaces analogiques. (V)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
•	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>RÉALISATION (E, F, G, H et P)</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>
• Mettre en pratique les lois de l'électricité appliquées à un circuit électronique alimenté en courant alternatif. (E, F, P)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
• Réaliser et dépanner des interfaces numériques. (F, I)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
• Réaliser et dépanner des interfaces analogiques. (E, I)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
• Appliquer des procédures de montage pour un circuit de moyenne envergure. (G)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
• Réaliser des montages électroniques comprenant un microcontrôleur communiquant avec des circuits mis en réseau multipoint à protocole simple. (G, H)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
•	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>DÉPANNAGE (J)</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>
• Concevoir et modifier des programmes structurés écrit en assembleur. (J)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
• Appliquer des procédures de dépannage pour un circuit de moyenne envergure. (J)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
• Dépanner des circuits de moyenne envergure. (J)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
•	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>INTÉGRATION (Q, U et T)</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>
• Concevoir et modifier des programmes structurés écrit en assembleur. (Q)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
•	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

## ANNEXE 1

## Schéma de la carte Driver PWM

On veut réaliser un PCB qui sera placé au dessus de la carte DRIVERS servomoteurs.

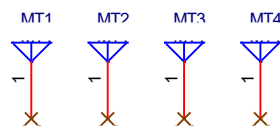
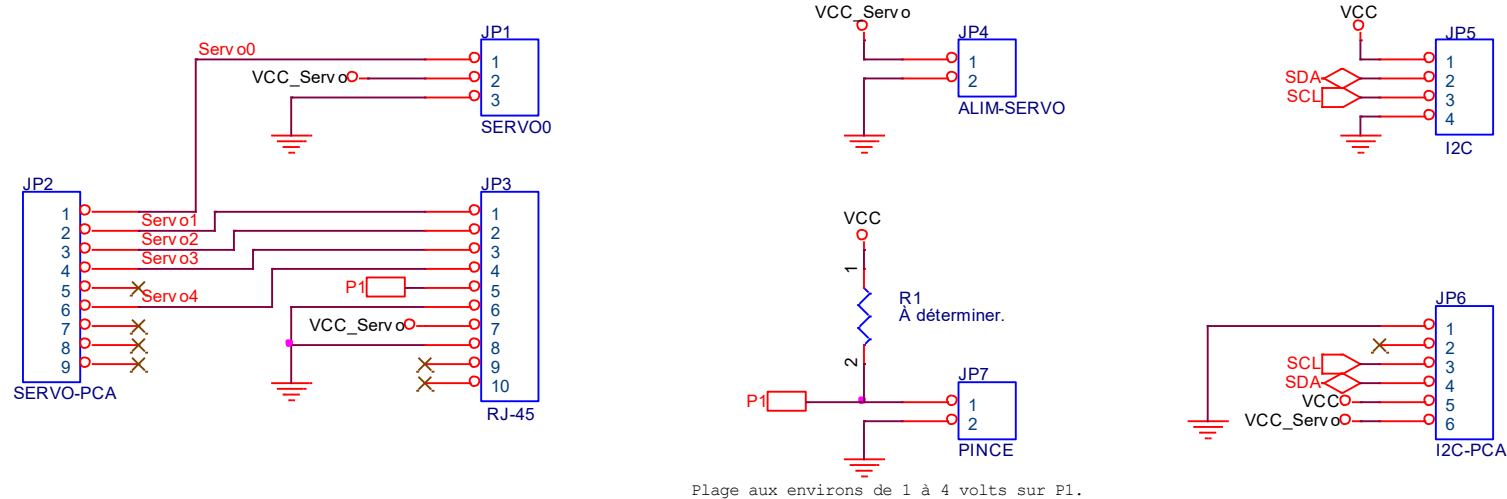
- Deux connecteurs toucheront directement à la DRIVE, Le connecteur JP2 qui sera branché aux signaux PWM des servomoteurs 0 @ 4. (connecteur de 9 broches pour plus de solidité)
- Et le connecteur JP6 pour le I2C et l'alimentation.

## Tâches:

- Modifiez le nom du fichier pour avoir NoEquipeProjS3.DSN. Mettez à jour la cartouche.
- Faites le DRC pour vérifier votre schéma.
- Générez le Wirelist et vérifiez l'ensemble des connexions.
- Générez le Netlist pour passer à Layout.
- Chercher une disposition qui vous permettra d'avoir un PCB assez petit et sans trop de via.
- Faites le PCB avec les contraintes suivantes:
  - Les nets sont à 20 mils.
  - Les grilles sont à 25, 50, 25 et 25.
  - On peut router seulement sur les couches TOP et BOT.
  - Le spacing sera de 12 mils partout.
  - Le circuit doit être fait sur la LPKF. Donc des obstacles devront être mis pour certaines pièces (nommez les obstacles et attachez-les aux empreintes).
  - Il devra y avoir un ground plane attaché au GND sur le TOP et sur le BOT.
  - Pour les étiquettes sur le silkscreen, seulement deux orientations sont permises (Line Width 7, Text Height 60).
  - Placer sur la couche TOP vos initiales (Line Width 10, Text Height 80).
  - Indiquer sur la couche TOP, quelques signaux importants pour vous repérer.
  - Les trous de perçage doivent être adaptés à chaque pièce (valeur standard).
- Lorsque le PCB vous semble OK:
  - Imprimez TOP et BOT avec les trous visibles. (KDHO)
  - Vérifiez si on peut le brancher sur la carte DRIVERS et assurez-vous que les trous de montage et les deux connecteurs soient bien alignés.
  - Faites vérifier par le prof.

## Générez:

- Les fichiers Gerber et le fichier .LMD (NoEquipeProjS3.LMD) pour la LPKF.



Daniel Côté /Alain Champagne		
Title CARTE INTERFACE PCA9685 / BRAS		
Size A	Document Number CARTE-PROJ-S3-DRIVERS.OPJ	Rev 1
Date:	Tuesday, November 13, 2018	Sheet 1 of 1