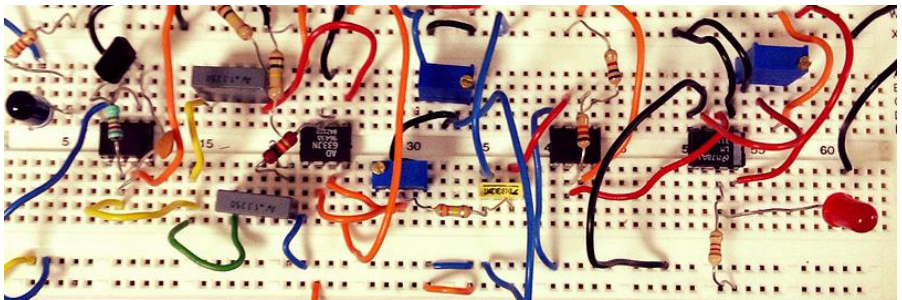


Électronique embarquée 1A S6 - EITI



F. BERNARD – J. VILLEMEJANE

Cycle Ingénieur - 1ère année - Palaiseau
Année universitaire 2017-2018
Version du 15 janvier 2018

Électronique TP

Calendrier	v
Règles de fonctionnement	vii
Modalités & Évaluation	ix
Thème 4 - Systèmes embarqués	1
.1 En partant de l'application	3
.2 Mesure de vitesse	15
.3 Contrôle de vitesse	21
.4 Utilisations avancées	27

Calendrier

Semaine	Date	Horaire	Groupe 1	Groupe 2	Groupe 3	Groupe 4	Groupe 5
2	lun. 8 janv.	8h30	MC 1				
	mar. 9 janv.	8h30				MC 1	
	mer. 10 janv.	8h30		MC 1			
	jeu. 11 janv.			Remise des synthèses AS		Remise des synthèses AS	
3	lun. 15 janv.	8h30	MC 2				
	mar. 16 janv.	8h30				MC 2	
	mer. 17 janv.	8h30		MC 2			
4	lun. 22 janv.	SITES					
5	lun. 29 janv.	8h30	MC 3 + exam				
	mar. 30 janv.	8h30			MC 1		
	mer. 31 janv.	8h30					MC 1
6	lun. 5 févr.	8h30	MC 4 + exam				
	mar. 6 févr.	8h30			MC 2		
	mer. 7 févr.	8h30					MC 2
7	lun. 12 févr.	8h30					MC 3 + exam
	mar. 13 févr.	8h30		MC 3 + exam	MC 3 + exam		
	mer. 14 févr.	8h30					
	jeu. 15 févr.		Remise des synthèses Th4				
8	lun. 19 févr.	VACANCES					
9	lun. 26 févr.	8h30					MC 4 + exam
	mar. 27 févr.	8h30				MC 3 + exam	
	mer. 28 févr.	8h30		MC 4 + exam			
10	lun. 5 mars	8h30			MC 4 + exam		
	mar. 6 mars	8h30				MC 4 + exam	
	mer. 7 mars	8h30					Remise des synthèses Th4
	jeu. 8 mars			Remise des synthèses Th4			Remise des synthèses Th4
11	jeu. 15 mars			Remise des synthèses Th4	Remise des synthèses Th4	Remise des synthèses Th4	

Version du 15 janvier 2018.

Il sera mis à jour sur le site EITI / Julien VILLEMEJANE

Absences

La présence des étudiant·e·s à toutes les séances de travaux pratiques prévues à l'emploi du temps est obligatoire et impérative. En cas de difficulté majeure, **si un membre d'un binôme est toutefois absent, l'autre doit venir à la séance et faire le TP**. Et, en Optique, chacun des membres du binôme rendra un compte-rendu individuel.

Absence excusée. Justificatif Le justificatif d'absence doit être déposé au secrétariat, les élèves concerné·e·s doivent aussi prévenir directement les responsables du LEnsE du motif de l'absence (à l'avance, si l'absence est prévisible).

Absence excusée. Rattrapage L'élève doit impérativement prendre contact avec les enseignant·e·s de TP pour étudier la possibilité de rattrapage (suivant la disponibilité des enseignant·e·s, du matériel et des salles). L'élève rattrape alors le TP et :

En optique, l'élève rédige un CR qui sera noté. S'il n'est pas possible de trouver une date de rattrapage suite à une impossibilité du service des TP, le TP ne sera ni rattrapé ni noté (la moyenne sera faite sur les notes restantes). Ce TP restera néanmoins au programme de l'examen et l'étudiant·e pourra être interrogé·e sur ce TP lors de l'examen de TP.

En ETI et ProTIS, la synthèse du thème concerné, rédigée par le binôme, devra contenir des résultats des deux séances individuelles (la séance normale et celle de rattrapage).

Si l'élève refuse la date de rattrapage proposée, il ou elle sera considéré·e comme absent·e non excusé·e.

Absence non excusée Toute absence non justifiée entraîne :

En optique, un zéro pour la séance et l'impossibilité de travailler sur ce TP avant la période de révision. En cas d'absences répétées, le responsable d'année interdira à l'étudiant·e de passer l'examen en fin d'année.

En ETI et ProTIS, un zéro pour la note de synthèse concernée.

Retards

Aucun retard n'est acceptable et en cas de retard important (ou de retards fréquents) d'un·e étudiant·e, celui-ci ou celle-ci se verra refuser l'accès au laboratoire. Les conséquences en seront identiques à celles d'une absence non excusée (voir plus haut).

Plagiats

Le plagiat est le fait de s'approprier un texte ou partie de texte, image, photo, données... réalisé par quelqu'un d'autre sans préciser qu'il ne s'agit pas de son travail personnel. On plagie quand on ne cite pas l'auteur des sources que l'on utilise. Exemples de plagiat :

- Copier textuellement un passage d'un livre ou d'une page Web sans le mettre entre guillemets et/ou sans en mentionner la source.
- Insérer dans un travail des images, des graphiques provenant de sources externes (hors énoncé du TP) sans en indiquer la provenance.
- Utiliser le travail d'un·e autre élève et le présenter comme le sien (et ce, même si cette personne a donné son accord !).
- Résumer l'idée originale d'un auteur en l'exprimant dans ses propres mots, mais en omettant d'en indiquer la source.
- Traduire partiellement ou totalement un texte sans en mentionner la provenance.

Tout binôme convaincu de plagiat dans un compte-rendu ou une synthèse de TP se verra attribuer la note de 0/20 à ce TP ou cette synthèse et encourt les sanctions disciplinaires prévues au règlement intérieur.

Respect du matériel et des locaux

Le LEnsE met à votre disposition une très grande quantité de matériel scientifique.

Ces matériels sont très fragiles, sensibles à la poussière, aux traces de doigts, aux rayures, etc. Merci d'en prendre le plus grand soin.

Il est donc formellement interdit d'apporter de la nourriture ou des boissons dans l'ensemble du service (couloirs compris). Merci de veiller aussi à laisser les locaux particulièrement propres (si vos chaussures sont sales, retirez-les et laissez-les à l'entrée !)

Pour toute demande d'accès en dehors des séances de TP, vous devez impérativement (et à l'avance) vous adresser au responsable technique du LEnsE, Thierry AVIGNON ou à Cédric LEJEUNE (bureau S1.18).

Au cours de chaque semestre, deux notes sanctionnent votre travail :

- Une **note par binôme** d'évaluation du travail de synthèse,
- Une **note individuelle** d'examen pratique.

Les notes de travaux pratiques d'ETI du semestre 5 sont prises en compte dans l'unité d'enseignement *Traitement de l'information 1* (5N-019-SCI | 8 ECTS). La note de synthèse contribue à 20% à la note finale de l'unité d'enseignement, celle de l'examen individuel contribue à 15%.

Pour le semestre 6, dans l'unité d'enseignement *Traitement de l'information 2* (6N-058-SCI | 4 ECTS), l'examen contribue à hauteur de 20% de la note, la synthèse à hauteur de 30%.

1. Cahier de manipulation

Il est indispensable que vous teniez à jour un *cahier de manipulation* par binôme. Ce cahier est la mémoire de tous les circuits et les mesures que vous avez réalisés. Il vous sera utile pour rédiger les synthèses et pour les séances de TP suivantes, y compris au deuxième semestre ou lors des examens.

Les enseignant·e·s s'assureront que vous disposez bien d'un tel cahier à chaque séance même si il n'est pas évalué, ce document reste un document interne au binôme.

Ce cahier devra prendre la forme d'un espace de travail partagé sur le *nuage* de l'Institut d'Optique (<https://cloud.institutoptique.fr>).

Il est important est que chaque membre du binôme puisse accéder à l'ensemble des documents lors de chaque séance.

2. Synthèses

Nous attendons de vous dans cette synthèse une véritable analyse des résultats de mesures relevés ou observés en cours de séance.

2.1. Objectifs et contraintes

Pour rédiger une synthèse, il faut toujours partir des objectifs visés et des contraintes imposées, contraintes sur le format, le contenu et aussi sur la date de remise.

Les objectifs de ce travail de synthèse sont :

1. Améliorer votre compréhension des concepts et de vous approprier les savoirs-faire (pouvoir dire : ça, je sais faire !),
2. vous entraîner à la présentation scientifique, élément qui fait partie à part entière de toute formation scientifique,
3. permettre aux enseignant-e-s de suivre la progression dans les apprentissages et d'évaluer par une note le travail effectué et les connaissances acquises.

Des conseils : ce sont les deux premiers objectifs qui doivent être gardés en tête lors de la rédaction, la note qui en découle n'est qu'une conséquence (heureuse !) de leur réussite ! Plus précisément :

1. **il faut rédiger la synthèse en imaginant que l'on s'adresse non pas à l'enseignant-e mais à une tierce personne** (un-e élève à l'autre bout du monde par exemple) qui souhaite faire les mêmes types d'expériences mais avec un matériel différent.
2. **il faut avoir l'ambition d'expliquer et non pas seulement de décrire.**

Les contraintes de format sont fixées à :

- 4 pages maximum pour chacun des thèmes traités au premier semestre : *Outils de l'électronique, Photodétection et Analyseur de Spectre*,
- 8 pages maximum pour le thème *Systèmes embarqués* du deuxième semestre,

au format .pdf, remis après la dernière séance du thème selon le calendrier de la page v.

La contrainte de pages maximum donne des indications du niveau de détails attendu pour les explications, forcément ici limité. La synthèse est donc plus courte que l'ensemble des notes et relevés de mesure pris en séance, le délai donne le temps de trier, organiser et commenter celles-ci.

La problématique à traiter est imposée, elle ne reprend pas forcément l'ensemble des notions abordées, elle n'est divulguée qu'à la fin de la dernière séance.

2.2. Contenu. Structure. Mise en forme.

Peut figurer dans une synthèse tout ce qui peut aider à la réalisation et à la compréhension des phénomènes étudiés : circuit, relevés de mesure, courbes et surtout commentaires. Présenter des erreurs que vous avez commises et la correction que vous avez apportée est aussi tout à fait acceptable !

Ne doit pas figurer une courbe ou une image sans légende, le brochage d'un composant, la couleur de la table, etc.

Toute image ou courbe doit être accompagnée d'une légende (titre, axes horizontal et vertical avec unités) et doit être citée dans le corps du texte (s'il n'y a rien à en dire, ce n'est sans doute pas la peine de la faire figurer!).

Structure du document Le document doit impérativement comporter :

- vos noms, avec votre numéro de binôme,
- le titre,
- une ou deux phrases d'introduction,
- une conclusion.

et il doit être paginé (numéros de page/ nombre total de pages).

La contrainte de format doit être suivie! Suivre les contraintes n'est pas forcément un frein à la créativité, au contraire, cela oblige parfois à explorer d'autres façons de faire, moins évidentes mais plus enrichissantes!

Quelques conseils dans l'utilisation d'un traitement de texte (Microsoft Word, Libre Office Writer ou LateX ou ...).

- La taille de la police doit être au minimum de 12 pts,
- les marges peuvent être réduites par rapport à celles définies par défaut par le logiciel,
- l'organisation sous deux colonnes peut permettre d'avoir un texte plus dense,
- utiliser le correcteur orthographique,
- placer des figures dans un tableau de 2 lignes, la 2ème ligne contenant le titre, peut faciliter la mise en page.

Exporter au format .pdf

Word : Menu Fichier / Enregistrer sous ... et choisir le format .pdf dans la fenêtre suivante.

Writer: Menu Fichier / Exporter au format pdf

Dépôt Merci de respecter les consignes suivantes :

- Certifiez l'originalité de votre travail en faisant figurer la mention : *Nous attestons que ce travail est original, que nous citons en référence toutes les sources utilisées et qu'il ne comporte pas de plagiat.*

- **Vérifiez que vos noms et le numéro de votre binôme figurent sur la première page de votre compte-rendu avant de la transformer en .pdf.**
- Assurez-vous que le nombre maximum de pages est respecté.
- Renommez le fichier .pdf selon le format :
G5B12MonNomEtCeluiDeMonBinomeSyntheseAS.pdf pour le binôme 12 du groupe 5.

Les synthèses doivent être déposées sur le site Libres Savoirs aux dates précisées dans le calendrier (page v).

Attention : un point de moins par jour de retard !

2.3. En résumé : checklist

- Les conditions expérimentales des résultats "affichés" sont elles définies ?
- Les mesures réalisées sont-elles analysées ?
- Les figures sont-elles analysées ? Si ce n'est pas le cas, sont-elle utiles ?
- Les figures sont-elles numérotées ?
- Les figures ont-elles des titres ?
- Les axes des figures sont-ils renseignés ?
- Les noms des auteurs sont-ils écrits ?
- Les pages sont-elles numérotées ?
- Le fichier porte-t-il un nom du type :
G5B12MonNomEtCeluiDeMonBinomeSyntheseAS.pdf ?
- Le fichier est-il au format PDF ?

2.4. Évaluation des synthèses

Une synthèse de qualité répond à la problématique posée sous la forme d'un véritable article scientifique.

Elle répond à la problématique posée si :

- les circuits/schémas/ relevés de mesures donnent des éléments pertinents en réponse à la problématique.
- les interprétations des résultats sont correctes.

La forme est de bonne qualité si :

- le plan est apparent et pertinent,
- les figures sont correctement présentées (axes / légendes/ unités) et citées dans le texte,
- elle fournit des données en quantité raisonnable (pas toutes les mesures, mais en nombre suffisant)
- elle suit les consignes de format (noms, plagiat, nombre de pages, introduction, etc.)

Une échelle de notation indicative est la suivante :

- 0/20 Note attribuée en cas d'absence non justifiée ou de plagiat.
- 5/20 Note sanction pour un rendu indigent (une page sans mise en forme ...c'est arrivé !)
- 8/20 Consignes non respectées. Echelles des graphes systématiquement manquantes, conditions expérimentales non précises....
- 11/20 Des informations sont données, présentées correctement mais l'exploitation des mesures est incorrecte.
- 15/20 Démarche claire avec quelques erreurs d'interprétations. Présentation de bonne qualité.
- 20/20 Document du niveau d'un article scientifique. Toutes les mesures sont correctes, bien expliquées et exploitées pour répondre à la problématique posée.

3. Cartes conceptuelles

Les cartes conceptuelles demandées ont pour contraintes (outre la question à laquelle elles doivent répondre) d'être lisibles après impression (couleur) sur une feuille A4.

Quelques indications Une carte conceptuelle est de bonne qualité si elle répond à la problématique posée avec

- un choix de "boîtes-concepts" pertinent et en nombre raisonnable (entre 5 et 15)
- des liens entre les concepts utilisant des verbes d'action, et donnant une information correcte
- une organisation graphique (disposition, couleur des boîtes, sens de lecture multiples) donnant du sens,
- une exploitation des mesures réalisées pour argumenter.

Évaluation Une échelle de notation indicative :

Boîtes-concepts :

- A** Intitulés cohérents avec la problématique et en nombre raisonnable
- B** nombre ou intitulé des boîtes mal choisi
- C** nombre et intitulé des boîtes mal choisi

Liens :

- A** tous présents (avec verbe) et tous corrects
- B** quelques liens manquants ou incorrects

C Peu de liens utiles ou justes

Structure :

A Utilisation de codes de couleur et de forme clairs et utiles, lecture facile de la carte. Répond à la problématique.

B De bonnes idées mais ensemble pas abouti.

C Uniforme et/ou illisible.

Illustrations :

A Des résultats de mesures et des schémas de circuits illustrent la carte et aident à comprendre.

B Peu d'illustrations utiles.

C Aucun résultat ou schéma.

4. Examen individuel

Un examen pratique a lieu chaque semestre. Il dure une heure et teste vos capacités expérimentales :

- maîtrise du matériel de laboratoire,
- protocoles et précautions pour des mesures simples,
- choix raisonné de composants.

Il n'est pas demandé de compte-rendu de manipulation. Vous avez accès aux synthèses et à votre cahier de manipulation.

Vous serez convoqué-e-s en demi-groupe selon le calendrier de la page v.

Une échelle de notation indicative est la suivante :

- | | |
|-------|---|
| 0/20 | Note attribuée en cas d'absence non justifiée |
| 5/20 | Pas de circuit ni de mesure réalisé, même avec aide. |
| 8/20 | Une mesure réalisée avec aide. |
| 11/20 | Un circuit et/ou mesure réalisé(es), avec aide. Quelques erreurs dans les explications. |
| 15/20 | Circuits et mesures réalisés de manière autonome. Quelques erreurs dans les explications. |
| 20/20 | Ensemble des circuits et mesures réalisés de manière autonome dans le temps imparti. Explications pertinentes données à l'oral. |

5. Points de pénalité

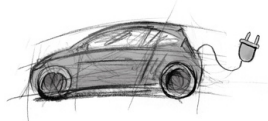
Des points de pénalité peuvent être attribués sur les notes de synthèses aux binômes n'ayant pas rangés leur poste de travail à l'issue des séances de TP.

1A S6 - EITI - Thème 4

Systemes embarqués

A travers ce thème, vous allez découvrir le monde des **systemes embarqués** et des **microcontrôleurs**.

Pour cela, vous allez apprendre à **piloter un moteur à courant continu** (organe mécanique de base dans tout système mobile), incluant la mise en œuvre d'un microcontrôleur, d'un étage de puissance et d'un afficheur LCD.



Objectifs

- Utiliser et optimiser une application simple d'électronique embarquée sur un microcontrôleur :
 - Utiliser les entrées/sorties numériques ;
 - Utiliser les entrées analogiques ;
 - Mettre en œuvre les interruptions externes ;
 - Mettre en œuvre les timers matériels ;
- Réaliser une interface homme-machine simple pour piloter un système industriel.

Préparation

Les différents énoncés de ce thème possèdent des questions notées **Px** à préparer avant chaque séance.

Rendu

Carte conceptuelle (*Mind Map*) et synthèse dont les sujets vous seront divulgués lors de la quatrième séance.

Documentation

Doc16F5013 *PIC16F1503 Data Sheet*. Microchip. Disponible sur S:\TP EITI\Theme4.

TutoMPLABX *Tutoriel du logiciel MPLABX*. Disponible sur chaque table et sur Libres Savoirs.

DocMPLABX *MPLABX IDE User's Guide*. Microchip. Disponible sur Libres Savoirs.

DocXC8 *XC8 User's Guide*. Microchip. Disponible sur Libres Savoirs.

Tutoriaux

Pour vous aider sur certains points, des **tutoriels** ont été mis en place. Vous les trouverez en ligne à l'adresse suivante :

<http://hebergement.u-psud.fr/villemejane/eiti/> dans la rubrique :

Tutoriaux > Microcontrôleurs
 > Microcontrôleurs industriels
 > Microcontrôleurs MICROCHIP PIC16F.

Vous pouvez également trouver quelques informations dans les tutoriels de 2A, sur le même site, mais dans la rubrique :

ProTIS > Tutoriels "A la carte".

Carte d'étude

Dans le cadre de ces TP, vous serez amenés à utiliser une carte de développement réalisée par le LEnsE, dont voici le brochage.

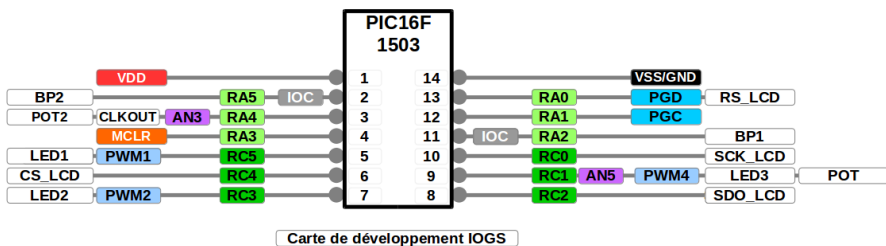


FIGURE 1 – Brochage du PIC16F1503 de la carte d'étude

Le schéma complet est fourni dans le tutoriel MPLABX, ainsi qu'en figure .1.5, page 9.

En partant de l'application

Cette première séance de TP est consacrée à la **découverte des moteurs à courant continu** qui équipent un grand nombre d'applications industrielles et "grand public" : transports électriques, chaînes de production, drones...



Dans une première partie, nous nous intéresserons au **fonctionnement de ces moteurs** afin de découvrir la meilleure façon de les **contrôler**.

Dans une seconde partie, nous prendrons en main un microcontrôleur de type industriel, organe servant essentiellement au **contrôle en temps réel** d'un système via des **broches de communication** spécifiques.

Sommaire

1	Étude du moteur à courant continu	5
1.1	Alimentation à tension constante	5
1.2	De l'optique pour mesurer la vitesse	6
1.3	Commande numérique	7
1.4	Commande en puissance	7
2	Découverte des microcontrôleurs PIC16F	9
2.1	Fichiers sources et ressources	10
2.2	Prise en main de MPLABX	10
2.3	Interface Homme-Machine	12

Après cette séance, vous serez capables de :

- identifier les paramètres importants sur un **moteur à courant continu**,
- alimenter un **moteur à courant continu**,
- utiliser la suite logicielle MPLABX,
- générer un programme simple pour un **microcontrôleur** de la famille des PIC
- configurer et utiliser les **entrées/sorties numériques** de ce microcontrôleur
- mettre en microcontrôleur une **interface homme-machine** simple

1. Étude du moteur à courant continu

Nous allons, avant tout, nous intéresser à la charge que nous allons vouloir contrôler : un moteur électrique à courant continu (DC motor) de la société PHILIPS/AIRPAX. Sa référence est 9904-120-52405. Il est distribué par le revendeur Radiospares sous la référence 336-337.

ATTENTION

La tension appliquée V_{mot} , entre les broches M+ et M- du moteur (broches jaune et verte), ne devra pas dépasser **6 V**.

1.1. Alimentation à tension constante

On se propose d'alimenter cette charge à l'aide d'une alimentation continue.

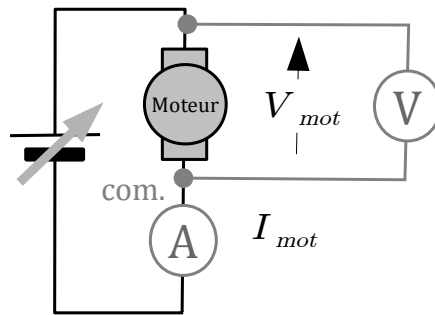


FIGURE .1.1 – Câblage du moteur à courant continu et d'un multimètre

M ⇨ Alimenter le moteur à l'aide d'une alimentation continue fixée à $V_{mot} = 5 \text{ V}$.

M ⇨ Placer le multimètre afin de pouvoir relever à la fois la tension aux bornes du moteur et le courant le traversant (voir figure .1.1).

M ⇨ Relever la valeur du courant.

M ⇨ Faire varier la tension d'alimentation du moteur entre 0 et 6 V.

Q1 Que se passe-t-il lorsque la tension aux bornes du moteur change ?

1.2. De l'optique pour mesurer la vitesse

Une fourche optique placée sur la maquette du moteur a pour référence HOA0901 Transmissive encoder sensor de la société Honeywell.

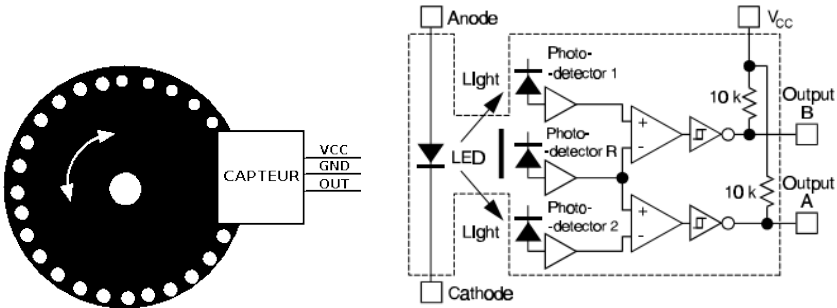


FIGURE .1.2 – Principe et schéma interne de la fourche optique HOA0901

Seule la sortie A de la fourche associée au moteur est connectée.

M \rightsquigarrow Alimenter cette fourche optique à l'aide d'une tension continue de 5 V.

M \rightsquigarrow Visualiser et relever la tension de sortie de cette fourche optique.

Q2 Comment calcule-t-on la vitesse de rotation du moteur Ω_{mot} ?

M \rightsquigarrow Relever la caractéristique $\Omega_{\text{mot}} = f(V_{\text{mot}})$ pour des tensions d'alimentation du moteur V_{mot} comprises entre 0 et 5 V.

Q3 Comment peut-on alors faire varier la vitesse de rotation d'un moteur à courant continu ?

1.3. Commande numérique

Une autre solution pour piloter ce type de charge, ayant un temps de réponse long (pour un moteur, essentiellement dû sa grande inertie), est d'utiliser des **signaux modulés en largeur d'impulsion** (voir figure .1.3).

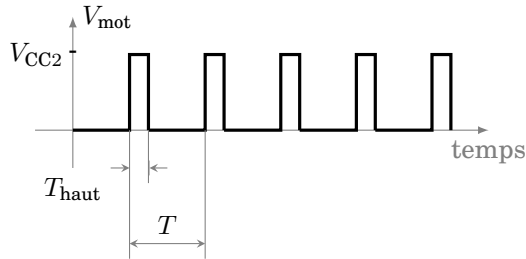


FIGURE .1.3 – Principe de la modulation de largeur d'impulsion

Ce principe a déjà été étudié dans les travaux pratiques du premier semestre.

M \rightsquigarrow Appliquer sur le moteur, à l'aide d'un GBF, un signal V_{mot} rectangulaire, de fréquence 100 Hz, de valeur minimale 0 V et de valeur maximale 5 V.

Q4 Pourquoi le moteur ne tourne-t-il pas ?

1.4. Commande en puissance

Pour pouvoir pallier à ce problème, nous allons utiliser un **transistor de puissance**, ici un **IRF540N** ou un **BS170** (voir avec l'enseignant), comme le montre le montage de la figure .1.4.

M \rightsquigarrow Câbler le montage de la figure .1.4, avec $R = 12 \text{ k}\Omega$ et $V_{CC} = 6 \text{ V}$.

M \rightsquigarrow Appliquer un signal carré de fréquence 100 Hz, de rapport cyclique 50% et d'amplitude 0 – 5 V sur la broche G du transistor IRF540N (ou BS170) à l'aide d'un GBF. On pourra utiliser la sortie SYNC du GBF.

M \rightsquigarrow Visualiser à l'oscilloscope la **sortie de la fourche optique** sur la voie 1 et la sortie du GBF sur la voie 2.

Q5 Que se passe-t-il sur le moteur ?

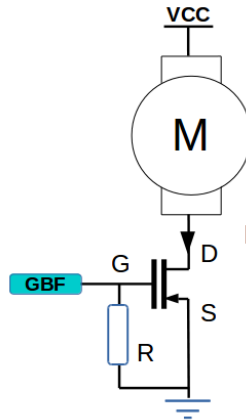


FIGURE .1.4 – Montage de puissance pour MCC

M \rightsquigarrow Modifier alors le rapport cyclique du signal appliqué sur l'entrée G du transistor.

Q6 Que se passe-t-il sur le moteur ?

2. Découverte des microcontrôleurs PIC16F

Afin de pouvoir contrôler la vitesse de rotation du moteur sans passer par des générateurs basse fréquence, nous allons nous intéresser à présent à la mise en œuvre d'un microcontrôleur. Pour cela nous allons utiliser une carte déjà pré-cablée contenant un microcontrôleur de type PIC16F1503, dont le schéma est donné ci-dessous en figure .1.5.

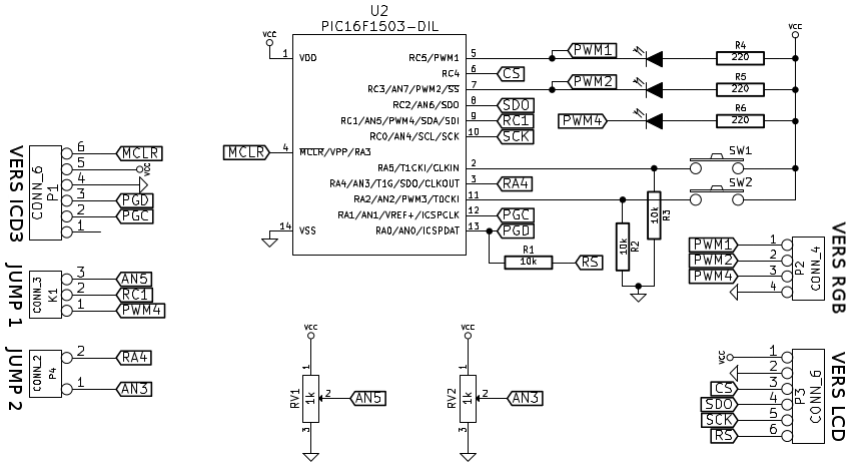


FIGURE .1.5 – Schéma de la carte d'étude développée par le LENSE

On rappelle également par la correspondance entre les broches du composant et l'application associée.

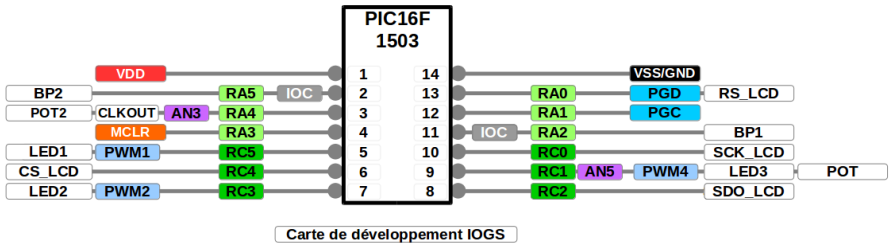


FIGURE .1.6 – Brochage du PIC16F1503 de la carte d'étude

2.1. Fichiers sources et ressources

REMARQUE

L'ensemble des fichiers utilisés lors de ce thème sont dans le répertoire `S:\TP EITI\Theme4\Sources`.

M ↪ Copier ce répertoire dans votre espace de travail sur `U:\`.

Ce répertoire contient :

- **Ressources** : répertoire contenant les bibliothèques que vous utiliserez par la suite :
 - `lcd.h` / `lcd.c` : écran LCD
 - `spi.h` / `spi.c` : liaison SPI
- **Sources** : répertoire contenant les fichiers sources que vous serez amenés à étudier au cours des différentes séances
- **PIC16F1503_vf.pdf** : documentation technique du PIC16F1503 dont sont tirées les pages mentionnées dans les fichiers
- **TutorielMPLABX_v100ans.pdf** : Tutoriel de prise en main du logiciel et de la carte d'étude (disponible également au format papier sur les paillasses).

2.2. Prise en main de MPLABX

A travers cette expérience, vous serez amené à découvrir l'environnement de développement **MPLAB X**, spécifique aux microcontrôleurs développés par Microchip, ainsi que la programmation en langage C de ce type de composant.

M ↪ Créer un projet intitulé `TP01.X` à l'aide de MPLABX pour un microcontrôleur de type PIC16F1503 (qui est le composant présent sur la carte d'étude), en suivant le **chapitre II.1** du tutoriel pour MPLABX. Nous utiliserons par la suite un programmeur de type ICD3.

M ↪ Connecter le programmeur ICD3 avec la carte de développement (sans oublier de l'alimenter), en suivant les **paragraphes I.3 et I.4.2** du tutoriel pour MPLABX.

M ↪ Ajouter le programme `00_IDE_main.c` au projet précédent, en suivant le **paragraphe II.1.2** du tutoriel pour MPLABX.

M ↪ Ajouter également le fichier `config.h` au projet précédent.

M ↪ Compiler le programme et le téléverser sur le microcontrôleur en suivant les **paragraphes II.3 et II.4** du tutoriel.

Q7 En analysant ce qu'il se passe sur la carte de développement lors de l'appui sur un bouton poussoir et en vous aidant des pages indiquées en commentaire du code ainsi que le **chapitre II.5** du tutoriel pour MPLABX, expliquer ce que fait ce programme. Quel est en particulier l'intérêt de la boucle `while(1)` ?

Q8 Comment peut-on écrire le code contenu dans la boucle `while(1)` plus simplement ?

M ⇨ Modifier ce programme pour allumer la seconde LED sur la carte à l'aide du second bouton-poussoir.

Q9 A quoi sert la fonction `DELAY` introduite dans ce code ?

M ⇨ Créer un fichier `01_DELAY_main.c` dans le projet précédent (en suivant le paragraphe II.1.2 du tutoriel MPLABX). Vous devrez également retirer le précédent fichier `.c` du projet : cliquer droit sur le fichier à retirer, puis sélectionner l'option `Remove from project`.

M ⇨ Concevoir un programme qui fasse clignoter la LED1 à une fréquence de 2 Hz.

M ⇨ Modifier ce programme pour que l'appui sur un bouton-poussoir modifie la fréquence de 2 Hz quand le bouton est relâché à 0.5 Hz quand le bouton est appuyé.

Q10 Le temps de réponse de ce système (lors de l'appui sur le bouton-poussoir) est-il constant ?

Q11 Que se passe-t-il si :

- vous retirez le connecteur de l'ICD3 ?
- vous débranchez puis rebranchez l'alimentation de la carte ?

2.3. Interface Homme-Machine

On souhaite à présent utiliser un écran LCD pour pouvoir réaliser une interface Homme-Machine simple. Pour cela, nous allons utiliser un écran LCD de type DOGM 163 de Electronic Assembly. Il comporte 3 lignes de 16 caractères, relié au microcontrôleur par l'intermédiaire d'une liaison SPI. Le codage utilisé est de l'ASCII, codé sur 8 bits.

M ↪ Connecter l'écran LCD à la carte de développement, en suivant le paragraphe I.4.2 du tutoriel MPLABX.

Une bibliothèque, composée des fichiers `lcd.h` et `lcd.c` (répertoire `S:/TP EITI/Theme4/Ressources`), vous est fournie et contient les fonctions suivantes :

void initLCD_DOG(void) pour initialiser le module LCD (liaison SPI et module d'affichage);

clearLCD(void) pour effacer l'ensemble de l'écran LCD;

setPosition(char ligne, char colonne) pour positionner le curseur sur l'écran LCD;

void writeLCD(char c) pour afficher un caractère à la position du curseur sur l'écran LCD;

void writeStrLCD(char c[], char ligne, char colonne) pour afficher une chaîne de caractère à une position particulière de l'écran LCD.

Cette bibliothèque nécessite également la bibliothèque `spi.h / spi.c` pour contrôler la liaison SPI entre le microcontrôleur et l'écran LCD.

M ↪ Ajouter le programme `02_LCD_main.c` à votre projet. Vous devrez également retirer le précédent fichier `.c` du projet : cliquer droit sur le fichier à retirer, puis sélectionner l'option `Remove from project`.

M ↪ Ajouter les fichiers associés aux deux bibliothèques mentionnées précédemment à votre projet.

M ↪ Compiler le programme et téléverser-le.

Q12 Qu'affiche l'écran LCD ? A quoi servent les différentes lignes de ce programme ?

Q13 La boucle `while(1)` est-elle nécessaire ?

M ↪ Ajouter une variable nommée `compteur` de type `int`, qui s'incrémente à chaque appui sur l'un des deux bouton-poussoirs.

M ↪ Afficher la valeur de cette variable sur l'une des lignes de l'écran LCD avec une chaîne de caractère du type : "CPT = val ". Vous pouvez utiliser la fonction `sprintf()` de la bibliothèque `stdio.h`.

Q14 A quelle vitesse s'incrémente cette variable? Comment peut-on modifier le programme précédent pour que chacune des valeurs s'affiche?

On souhaite détecter les changements de niveau (les fronts) sur le bouton. Pour la détection de front on propose un exemple de code sur le listing .1.1

Listing .1.1 – Programme d'exemple pour la détection de front

```
1 new_a = bouton ;
2     if((new_a == 1) && (old_a != new_a)){
3         c++;
4     }
5     old_a=new_a;
```

Q15 Cet exemple correspond-il à une détection de front montant ou descendant? Sur quelle variable?

M ↪ Modifier le programme pour détecter un front montant sur le bouton-poussoir que vous avez choisi.

Q16 Que se passe-t-il à présent?

M ↪ Connecter à présent la sortie de la fourche optique à l'entrée RA4 de la carte. Les broches à connecter sont indiquées sur la photographie de la figure .1.7.

M ↪ Modifier le programme pour qu'il incrémente la variable `compteur` à chaque front montant sur RA4.

Q17 Que se passe-t-il à présent?

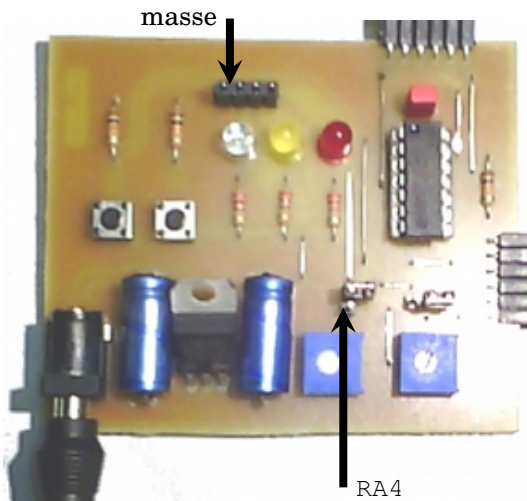


FIGURE .1.7 – Connexions sur la carte

Mesure de vitesse

Après avoir découvert le microcontrôleur et ses fonctionnalités de base (entrées/sorties numériques) et élaboré une **interface homme machine simple**, nous allons, durant cette séance, voir comment nous pouvons nous servir de ce système de base pour **mesurer la vitesse de rotation** du moteur.



Après cette séance, vous serez capable de :

- mettre en œuvre une **routine d'interruption**,
- configurer et utiliser les **interruptions externes**,
- paramétrer le module de **gestion du temps** (timer)

Sommaire

1	Mesure de la vitesse de rotation	16
2	Interruptions	16
3	Gestion du temps	18
4	Interruptions multiples	19
5	Mesure et affichage de la vitesse	20

1. Mesure de la vitesse de rotation

On se propose durant cette séance d'analyser les signaux fournis par la fourche optique associée au moteur à courant continu, étudié lors de la séance précédente, afin de récupérer la vitesse de rotation de l'arbre du moteur.

P1 A partir des signaux obtenus lors de la précédente séance, expliquer dans les grandes lignes l'algorithme que nous devons utiliser pour mesurer la vitesse de rotation du moteur et permettre de l'afficher sur l'écran LCD.

P2 Est-il possible de réaliser ce programme à l'aide des fonctionnalités déjà étudiées du microcontrôleur ? Expliquer.

2. Interruptions

Le mode de fonctionnement vu lors de la séance précédente, où l'on vient **scruter les entrées** dans la boucle infinie (mode *polling*) dans le programme `main` n'est pas efficace pour les programmes temps réel.

Afin de pouvoir **interagir plus rapidement** avec son environnement et prendre en compte des **événements extérieurs**, les microcontrôleurs sont dotés d'une capacité à interrompre l'exécution d'un programme pour se détourner vers une fonction particulière à exécuter lors de l'arrivée d'un événement extérieur. On appelle cela une **interruption**.

Pour cela, le microcontrôleur possède **plusieurs entrées particulières** qui permettent d'interrompre le programme principal. Plusieurs autres modules du microcontrôleur (*timers*, ADC¹...) peuvent également venir interrompre l'exécution du programme principal.

2.1. Interruptions externes

Nous allons nous intéresser dans cette partie aux broches notées **IOC** qui permettent d'interrompre le microcontrôleur via des événements externes (front montant ou/et descendant sur une broche du composant).

Nous allons plus particulièrement nous intéresser à la broche **RA2** sur le PIC16F1503, associée au bouton-poussoir **SW2** sur la carte d'étude.

P3 En vous aidant du schéma de la carte d'étude donné en figure .1.5, quelle est la valeur au repos appliquée sur la broche **RA2** ? A quoi sert la résistance R_2 associée ?

1. *Analog to Digital Converter* ou convertisseur analogique-numérique

2.2. Structure du programme

On se propose dans un premier temps d'analyser et de tester le programme contenu dans le fichier source `03_INT_main.c`. Il va nous permettre de voir quelle **structure** doit avoir un programme écrit **pour gérer des interruptions externes**.

M ↪ Créer un projet intitulé `TP02.X` à l'aide de MPLABX pour un microcontrôleur de type PIC16F1503, en suivant le chapitre II.1 du tutoriel pour MPLABX. Nous utiliserons par la suite un programmeur de type ICD3.

M ↪ Ajouter le fichier `03_INT_main.c` à ce nouveau projet.

M ↪ Compiler le programme et téléverser-le.

Q1 Que fait ce programme ? A quel moment est appelée la fonction `void prog_interruption(void)` ?

Cette fonction particulière est appelée **routine d'interruption**.

Q2 A quel rythme est pris en compte l'appui sur le bouton-poussoir connecté sur `RA2` ? Qu'est-ce qui est amélioré par rapport à la séance précédente ?

M ↪ Modifier le programme précédent pour qu'il incrémente une variable entière nommée `compteur` à chaque appui sur `SW2` et l'affiche sur l'écran LCD.

M ↪ Compiler le programme et téléverser-le.

M ↪ Modifier le programme précédent pour qu'il décrémente cette même variable à chaque appui sur `SW1` et l'affiche sur l'écran LCD.

Q3 L'appui sur les bouton-poussoirs est-il pris en compte instantanément ?

3. Gestion du temps

Un des principaux challenges des systèmes embarqués vient du fait qu'ils soient capables d'**exécuter leurs tâches dans un temps imparti**. On parle alors de systèmes "**temps réel**".

En dehors des fonctions de type `delay`, qui monopolisent le processeur, les microcontrôleurs possèdent des modules spécifiques de gestion du temps : les **timers**, qui génèrent des interruptions à intervalle régulier.

3.1. Principe du timer TMR0

On se propose ici d'étudier l'un de ces timers matériels, nommée *timer 0* ou TMR0. Le schéma de principe est donné en figure .2.1.

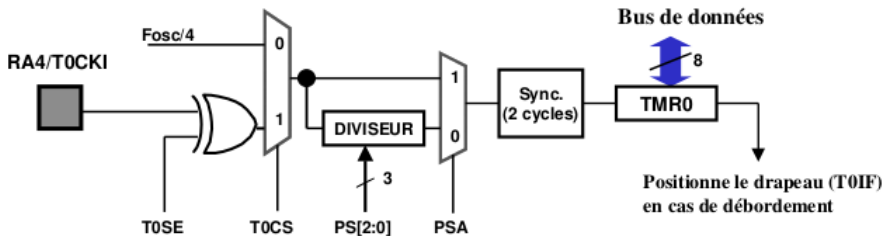


FIGURE .2.1 – Schéma du timer matériel TMR0

Le timer TMR0 est un **compteur 8 bits** qui s'incrémente à chaque front montant de l'horloge d'entrée et produit un **débordement** lorsqu'il repasse de sa valeur maximale (ici 255) à 0. Il génère alors une interruption détectable sur TMR0IF du registre INTCON.

L'horloge d'entrée de ce compteur est également paramétrable : on peut ainsi choisir l'oscillateur interne ou une entrée extérieure. On peut également utiliser un **pré-diviseur** pour réduire la fréquence en entrée du compteur TMR0, les chronogrammes de ces différents fonctionnements sont indiqués sur la figure .2.2.

La relation qui relie la fréquence des interruptions sur le timer TMR0 et la fréquence d'oscillation (dans le cas du choix de l'oscillateur interne) est la suivante :

$$f_{\text{TMR0}} = \frac{F_{\text{OSC}}}{4} \cdot \frac{1}{2^8 \cdot \text{PREDIV}}$$

où F_{OSC} est la fréquence choisie pour l'oscillateur interne (voir registre OSCCON) et PREDIV la valeur choisie pour le prédiviseur.

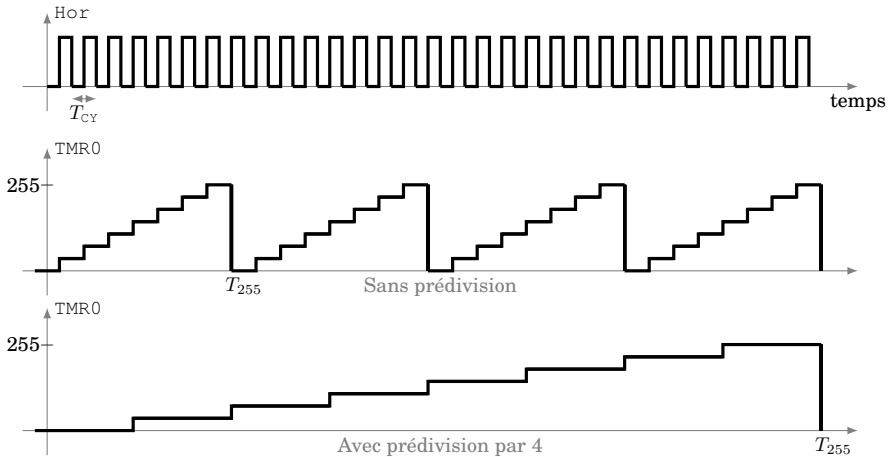


FIGURE .2.2 – Chronogrammes d’un timer avec ou sans prédivison de fréquence d’horloge. La période de l’horloge est notée T_{Cy} , elle correspond à 4 périodes de l’oscillateur interne (dans le cas du choix de l’oscillateur interne) .

M \rightsquigarrow Ajouter le fichier `04_TMR_main.c` au projet précédent.

M \rightsquigarrow Compiler le programme et téléverser-le.

M \rightsquigarrow Mesurer à l’aide d’un oscilloscope la fréquence de clignotement de la LED1.

Q4 Cette fréquence est-elle en adéquation avec celle configurée dans ce programme ?

Q5 Comment est modifiée la routine d’interruption par rapport au programme précédent ? (`03_INT_main.c`)

M \rightsquigarrow Modifier ce programme pour obtenir une fréquence de clignotement d’environ 100 Hz sur la LED1.

4. Interruptions multiples

Il est possible de combiner plusieurs types d’interruptions : externes, timers... dans une même application. On peut ainsi prendre en compte des événements extérieurs (appui sur des bouton-poussoirs) en même temps que répéter une tâche à intervalle régulier.

M ↗ A partir des exemples précédents, créer une application qui permettent de modifier la fréquence de clignotement de la LED2 par appui sur les bouton-poussoirs SW1 et SW2. L'appui sur SW1 permettra de faire clignoter la LED2 à la fréquence d'environ 100 Hz et l'appui sur SW2 à la fréquence d'environ 400 Hz.

M ↗ Compiler le programme et le téléverser.

M ↗ Tester le bon fonctionnement de votre application.

5. Mesure et affichage de la vitesse

L'objectif final de cette séance est de réaliser l'affichage de la vitesse de rotation du moteur à l'aide d'un écran LCD.

M ↗ En combinant les interruptions externes et le timer, concevoir un programme qui permet de compter un nombre d'impulsions sur l'entrée RA4 dans un temps donné et calibré.

Q6 Comment peut-on utiliser ce programme pour afficher la vitesse de rotation du moteur ?

M ↗ Afficher la vitesse de rotation du moteur en tour/min sur l'écran LCD.

Contrôle de vitesse

Le système que l'on cherche à concevoir doit permettre de contrôler la vitesse de rotation du moteur à l'aide du microcontrôleur, afin de s'affranchir des générateurs de signaux.

Pour cela, nous allons utiliser un potentiomètre dont nous prélevons la tension de sortie à l'aide d'une **entrée analogique** du microcontrôleur. Cette entrée analogique est ensuite associée à un convertisseur analogique-numérique de 10 bits que nous mettrons en œuvre.

La valeur convertie servira ensuite pour paramétrer le rapport cyclique d'un signal modulé en largeur d'impulsion, qui sera généré par le module **PWM** du microcontrôleur.

Après cette séance, vous serez capable de :

- acquérir un **signal analogique** et le traiter numériquement
 - paramétrer le module **PWM** pour générer des signaux modulés en largeur d'impulsion
-

Sommaire

1	Acquérir un signal analogique	22
2	Sorties modulées en largeur d'impulsion	23
3	Système complet	26

1. Acquérir un signal analogique

Afin de pouvoir utiliser le signal analogique sortant d'un potentiomètre, il est nécessaire de le convertir en numérique. Pour cela, le PIC16F1503 dispose d'un **convertisseur analogique-numérique** (CAN ou *ADC - Analogic-to-Digital Converter*) de 10 bits. Il est possible de sélectionner une des **8 entrées analogiques**, notées AN_x (de AN0 à AN7), pour être convertie par le CAN.

Sur la carte d'étude, seules les entrées AN3¹ et AN5² sont reliées à circuits à base de potentiomètres, dont la sortie peut évoluer entre 0 et 5 V.

P1 À partir de la documentation technique du microcontrôleur (section 15 et registres 15-1 à 15-7), déterminer quels sont les registres qui permettent de configurer le convertisseur analogique-numérique (voir aussi TD11 ETI, semestre 1).

M ⇨ Créer un projet intitulé TP03.X à l'aide de MPLABX pour un microcontrôleur de type PIC16F1503, en suivant le chapitre II.1 du tutoriel pour MPLABX. Nous utiliserons par la suite un programmeur de type ICD3.

M ⇨ Créer un fichier 05_ADC_main.c à ce nouveau projet.

On souhaite par la suite récupérer la tension présente sur le potentiomètre RV1 connecté sur la broche RC1 correspondant à l'entrée analogique AN5.

On se propose d'étudier la fonction suivante :

```

1 void initADC(void){
2     TRISCbits.TRISC1 = 1;
3     ANSELbits.ANSC1 = 1;
4
5     ADCON0bits.CHS = 5;
6     ADCON1bits.ADFM = 1;
7     ADCON1bits.ADPREF = 0b00;
8     ADCON1bits.ADCS = 0b011;
9     ADCON0bits.ADON = 1;
10 }
```

1. Jumper J2 connecté - voir Tutoriel MPLABX
2. Jumper J1 du côté de la sortie LCD

Q1 Expliquer ce que fait cette fonction.

M \rightsquigarrow Recopier la fonction `void initADC(void)` dans votre fichier source principal.

M \rightsquigarrow Ecrire un programme qui permette de convertir toutes les 100 ms la tension présente sur le potentiomètre et d'afficher la valeur convertie sur l'écran LCD.

M \rightsquigarrow Compiler le programme et téléverser-le.

Q2 Quelles sont les valeurs minimale et maximale affichées ?

M \rightsquigarrow Modifier le programme pour afficher la tension associée à cette valeur sur l'écran LCD.

2. Sorties modulées en largeur d'impulsion

Afin de pouvoir commander l'étage de puissance associée au moteur, nous allons utiliser le principe de modulation de largeur d'impulsions (ou PWM) disponible sur ces microcontrôleurs.

2.1. Principe de fonctionnement

La fonction PWM des PIC16F1503 est la combinaison de 2 périphériques :

- le timer TMR2 (commandé par l'horloge interne) ;
- un comparateur.

Un **signal rectangulaire de période fixée** par le registre PR2 est produit avec un **rapport cyclique** pouvant être réglé grâce aux registres PWMxDCH et PWMxDCL.

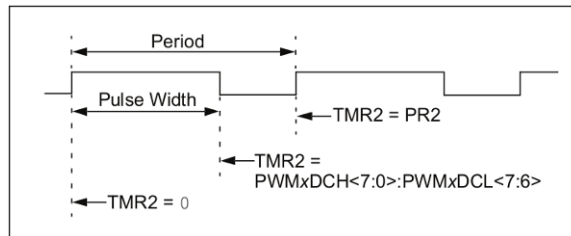


FIGURE .3.1 – Principe de la sortie modulée sur les microcontrôleurs PIC. (Source documentation constructeur)

Le **timer** va servir à générer un **signal de modulation** de forme triangulaire (compteur binaire) qui fixe la **fréquence** (ou période) du signal de sortie.

Le **comparateur** va en permanence comparer ce signal de modulation à une valeur de **référence** permettant de fixer le **rapport cyclique** du signal de sortie.

La période de sortie du module PWM est donnée par la formule suivante :

$$P_{\text{PWMx}} = (\text{PR2} + 1) \cdot \frac{4}{F_{\text{OSC}}} \cdot \text{PREDIV}_{\text{TMR2}}$$

Q3 Avec une fréquence d'oscillation F_{OSC} de 2 MHz, quelle doit être la valeur de prédivison à configurer sur le timer TMR2 ainsi que la valeur du registre PR2 pour obtenir une fréquence de sortie du PWM de 1 kHz ?

Le rapport cyclique est donné par la relation suivante :

$$\text{DC}_{\text{PWMx}} = \frac{\text{PWMxDCH} : \text{PWMxDCL}}{4 \cdot (\text{PR2} + 1)}$$

Q4 Quelle est la valeur maximale que peut prendre l'ensemble PWMxDCH : PWMxDCL pour avoir un rapport cyclique compris entre 0 et 1 ? Quelle doit être la valeur de PWMxDCH : PWMxDCL pour avoir un rapport cyclique de 0,5 ?

Pour modifier le rapport cyclique du module PWM1, on propose la fonction suivante, qui prend en paramètre une valeur comprise entre 0 et 100 :

```

1 void changeDC1(char val){
2     int temp = (val * 4 * (PR2 + 1)) / 100;
3     PWM1DCL = temp << 6;
4     PWM1DCH = temp >> 2;
5 }
```

Q5 Quelle est la valeur maximale que peut prendre la variable temp ? Que permettent de faire les lignes 3 et 4 ?

2.2. Mise en œuvre

On souhaite, dans un premier temps, configurer la sortie `PWM1` du composant. Pour cela, on utilise la fonction suivante :

```
1 void initPWM1(void){
2     TRISCbits.TRISC5 = 0;    // PWM1
3
4     PWM1CONbits.PWMIEN = 1;
5     PWM1CONbits.PWMIOE = 1;
6     PWM1CONbits.PWMIPOL = 0;
7
8     PR2 = 31;
9     changeDCPWM1(0);
10    T2CONbits.T2CKPS = 0b10;
11    T2CONbits.TMR2ON = 1;
12 }
```

Q6 À partir du chapitre **22.1** et des registres **22-1**, **22-2** et **22-3**, expliquer ce que fait cette fonction. On supposera que la fréquence d'oscillation du microcontrôleur sera de 2 MHz (qu'il faudra penser à configurer...)

M ↪ Créer un fichier `06_PWM_main.c` que vous ajouterez à votre projet.

M ↪ Recopier les fonctions `void initPWM1(void)` et `void changeDC1(char val)` dans votre fichier source principal.

M ↪ Tester ces deux nouvelles fonctions en visualisant le signal de la sortie `PWM1` à l'aide d'un oscilloscope.

Q7 Comment peut-on faire pour obtenir un signal nul à l'aide de ce principe?

M ↪ Modifier le programme pour que le rapport cyclique du signal modulé sur `PWM1` soit commandé par la position du potentiomètre `RV1`.

M ↪ Tester votre application en vérifiant que le rapport cyclique du signal de la sortie `PWM1` soit bien modifié en fonction de la position du potentiomètre `RV1`.

2.3. Pilotage du moteur

Le système que nous venons de développer doit à présent permettre de piloter le moteur à courant continu vu dans la première séance de TP.

M ⇨ Recâbler le moteur et le composant IRF540N (voir séance 1 du thème 4 - figure .3.2).

M ⇨ Connecter la sortie PWM1 de la carte d'étude à l'entrée G du transistor IRF540N pilotant en puissance le moteur (voir figure suivante).

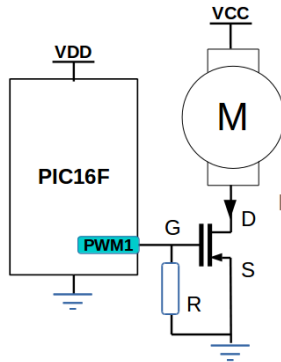


FIGURE .3.2 – Montage de puissance pour MCC avec un microcontrôleur

M ⇨ Tester votre application.

3. Système complet

On souhaite finalement pouvoir piloter la vitesse du moteur à l'aide d'une sortie modulée du microcontrôleur et afficher la vitesse de rotation sur l'écran LCD.

M ⇨ A l'aide des différents exemples réalisés au cours des trois premières séances de ce thème, proposer une application qui puisse remplir ce cahier des charges.

M ⇨ Tester votre application.

M ⇨ Ajouter la possibilité de commander l'allumage et l'extinction du moteur à l'aide d'un bouton-poussoir.

FELICITATIONS !!!

Vous venez de réaliser votre première application embarquée !

Utilisations avancées

Dans cette dernière séance, vous découvrirez, en fonction des projets que vous aurez choisi, des **possibilités plus avancées** avec les micro-contrôleurs : communication inter-systèmes, utilisation de capteurs numériques...

En fonction de votre projet, vous allez pouvoir choisir parmi les différents sujets suivants :

Sommaire

1	Moteur à courant continu / Pont en H	28
2	Liaison SPI / DAC externe	29
3	Liaison SPI / Mémoire externe	29
4	Accéléromètre analogique ADXL335	29
5	Liaison RS232 / Communication inter-système . . .	29

Certains de ces sujets sont uniquement disponibles en ligne :
<http://hebergement.u-psud.fr/villemeiane/eiti/> dans la rubrique :
Tutoriaux > Microcontrôleurs
> Microcontrôleurs industriels
> Microcontrôleurs MICROCHIP PIC16F.

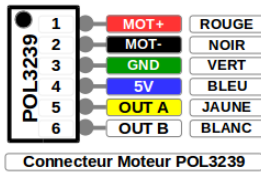
1. Moteur à courant continu / Pont en H

On se propose à présent de réaliser le pilotage d'un autre moteur à courant continu, de type `POLOLU POL3239`, vendu par la société `Lextronic` (référence : `POL3239`), dans les deux directions à l'aide d'un microcontrôleur et de ses sorties PWM.

Ce moteur possède un encodeur de **979,62 impulsions/révolution** et les caractéristiques suivantes :

- **Dimensions** : diamètre 25 mm / longueur 62 mm (arbre 4 mm avec méplat)
- **Poids** : 98 g
- **Réduction** : 1 :20,4
- **Alimentation** : 6 à 12 V
- **Vitesse à vide** : 370 rpm pour 12 V
- **Courant à vide** : 200 mA pour 12 V
- **Courant en charge** : 2,1 A pour 12 V
- **Couple** : 3 kg.cm pour 12 V

Le brochage du moteur est donné dans la figure suivante :



Plutôt que d'utiliser le montage .. du premier TP de ce thème, constitué d'un seul transistor de puissance (et ne permettant alors que le pilotage du moteur dans un seul sens de rotation), on s'intéresse à présent au contrôleur de puissance `L293DNE` (référence `1470423` chez Farnell). On appliquera sur ce composant les tensions suivantes : $V_{CC2} = 9\text{ V}$ (alimentation de puissance) et $V_{CC1} = 5\text{ V}$ (alimentation de commande - niveau TTL).

M ↔ Câbler le montage en utilisant les broches 1Y et 2Y vers chacune des broches de puissance du moteur (nommée MOT+ et MOT-).

M ↔ Connecter l'entrée 2A à la masse et l'entrée 1, 2EN à 5 V.

M ↔ Appliquer un signal carré de fréquence 100 Hz, de rapport cyclique 50% et d'amplitude 0–5 V sur la broche 1A du `L293DNE` à l'aide d'un GBF. On pourra utiliser la sortie SYNC du GBF.

M ↔ Visualiser à l'oscilloscope la sortie de la fourche optique sur la voie 1 et l'entrée 1A sur la voie 2.

Q1 Que se passe-t-il sur le moteur ?

M ↔ Modifier alors le rapport cyclique du signal appliqué sur l'entrée 1A.

Q2 Que se passe-t-il sur le moteur ?

M ↔ Inverser à présent les signaux sur les entrées 1A et 2A du L293DNE.

Q3 Dans quel sens tourne à présent le moteur ? Expliquez alors le principe de fonctionnement du L293DNE.

On va à présent appliquer des signaux PWM sur les entrées 1A et 2A à l'aide du microcontrôleur.

M ↔ Réaliser un programme qui génère deux signaux PWM (de même fréquence) dont les rapports cycliques seront contrôlés par un potentiomètre et la direction (fonctionnement de l'une ou de l'autre des sorties PWM) sera modifiée par l'appui sur un bouton-poussoir (ou 2 différents, selon vos envies).

M ↔ Vérifier vos signaux à l'oscilloscope, avant de les appliquer sur les entrées 1A et 2A du L293D.

2. Liaison SPI / DAC externe

3. Liaison SPI / Mémoire externe

4. Accéléromètre analogique ADXL335

5. Liaison RS232 / Communication inter-système