SII – PCSI / TP – Série n° 6

Commande prototype des systèmes numériques

Commande du vérin électrique

Objectifs

De nos jours, la plupart des systèmes automatisés sont pilotés par des micro-ordinateurs ou des processeurs numériques. L'objectif de cette série de TP est d'introduire les notions élémentaires nécessaires à l'étude et l'utilisation des systèmes numériques de contrôle et de commande. À l'issu de cette séquence, vous devrez être capable d'analyser, de modéliser et de programmer le comportement de tout ou partie d'un système ; notamment :

- identifier les paramètres d'entrée et de sortie d'un système numérique existant ;
- communiquer avec le système de commande afin de modifier la commande ;
- implémenter un algorithme de commande sur ordinateur ;
- mettre en place une chaîne fonctionnelle, de la commande du pré-actionneur à l'actionneur ;
- mettre en place une chaîne de mesure, du câblage du capteur à la maitrise de l'information acquise (fréquence d'acquisition, quantification, seuillage) ;
- mettre en place un asservissement avec une correction proportionnelle, si besoin avec saturation.

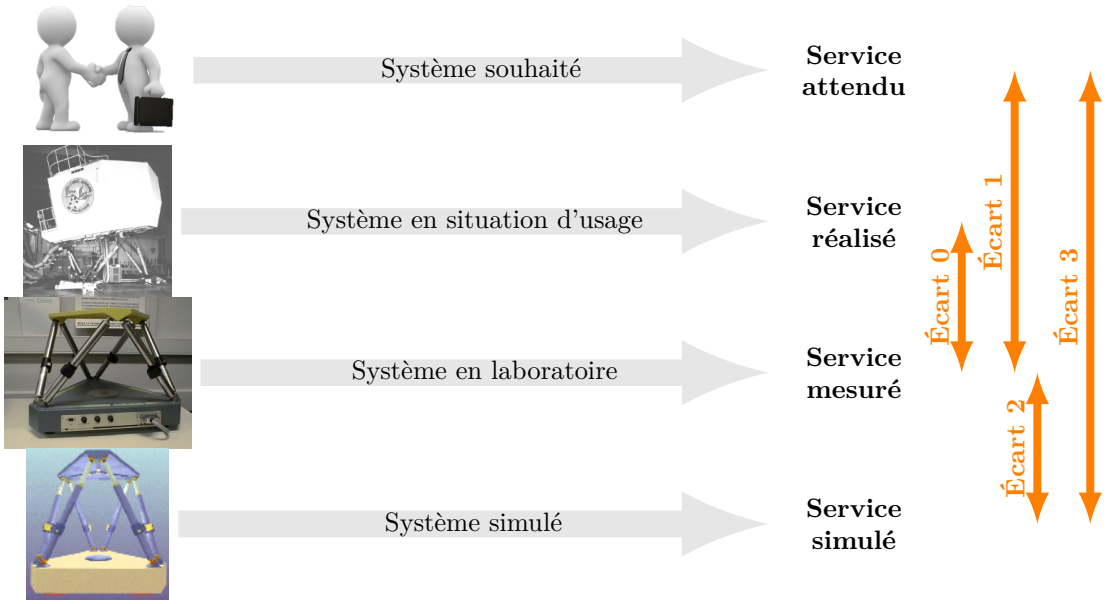


FIGURE 1 – Démarche de l'ingénieur centrée sur la mesure des écarts.

1 Introduction

L'objectif de ce TP est de mettre en place une commande d'asservissement de position du vérin électrique avec une carte pyboard. Afin de vous permettre de progresser le plus rapidement possible, la démarche suivante vous est proposée :

étape 1 – prise en main de la carte pyboard ;
étape 2 – commande directe du moteur ;
étape 3 – mise en place d'un asservissement de position.

1.1 Carte pyboard

La carte pyboard contient un microcontrôleur STM32F405RG (168 MHz) et propose :

— 24 entrées/sorties binaires (GPIO) dont 20 pouvant fournir un signal PWM ;
— 12 entrées analogiques converties sur 12 bits ;
— 2 sorties analogiques (non utilisées ici) ;
— des connecteurs pour liaisons séries (non utilisés ici) ;
— un lecteur de carte SD formatée en FAT (non utilisé ici) ;
— un accéléromètre (non utilisé ici) ;
— 4 LED et un bouton poussoir programmables.

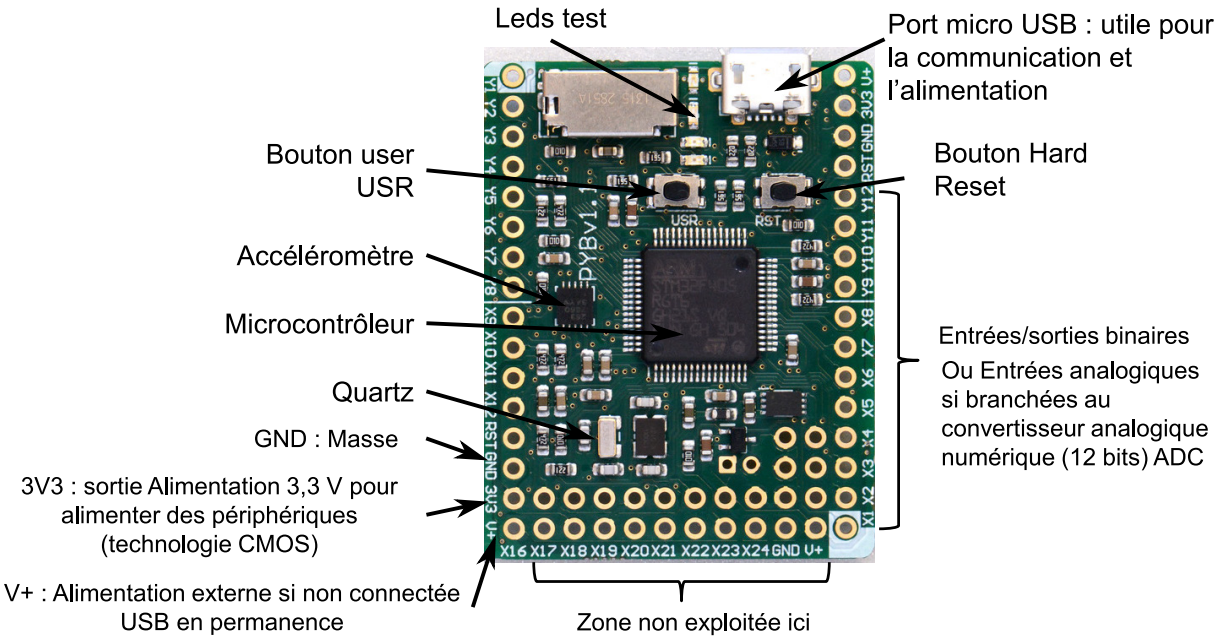


FIGURE 2 – Carte Pyboard.

La carte Pyboard possède son propre environnement appelé micropython. Vos programmes seront écrits sur votre ordinateur en python avec l'éditeur de votre choix, puis sauvegardés. Ils seront copiés sur la carte pyboard (comme sur une carte SD) à la place du fichier `main.py`. Il est possible d'interagir directement avec la carte à l'aide d'un terminal émulateur série (PuTTY avec Microsoft Windows ou screen avec GNU/Linux).

1.2 Connexion et LED

Connecter le port micro-USB à la carte pyboard puis le port USB sur le PC. Initialisez ensuite une connexion série avec la carte pyboard avec le logiciel PuTTY et l'aide fournie en annexe page 10. Tapez ensuite dans le *shell* :

```
import pyb
pyb.LED(1).on() # pour allumer la LED 1 (aussi 2, 3 et 4)
pyb.LED(1).off() # pour l'éteindre
```

ATTENTION

Vous devez impérativement réaliser tous vos montages hors tension et le faire vérifier par le professeur.

1.3 Lecture d'un potentiomètre

Vous allez dans cette partie lire les informations renvoyées par un potentiomètre. Vous devez réunir le matériel :

- 1 carte pyboard ;
- 1 potentiomètre de quelques kΩ ;
- 1 plaque de prototypage (pour montage sans soudure) ;
- des fils de prototypage.

FIGURE 3 – Potentiomètre de quelques kΩ.

Manipulation 1.
Fixer le potentiomètre sur la plaque de prototypage en prenant soin de mettre les broches sur 3 lignes différentes.
Connecter les deux fiches extérieures du potentiomètre aux voies GND et 3,3 V de la carte pyboard.
Connecter la fiche du milieu à la voie `X5`, une des entrées associées au ADC (*Analog Digital Converter*). La conversion analogique–numérique est réalisée sur 12 bits.

Déclarer ensuite cette entrée avec le code :

```
import pyb
pot = pyb.ADC('X5')
```

Pour lire la valeur, il suffit alors de faire :

```
pot.read()
```

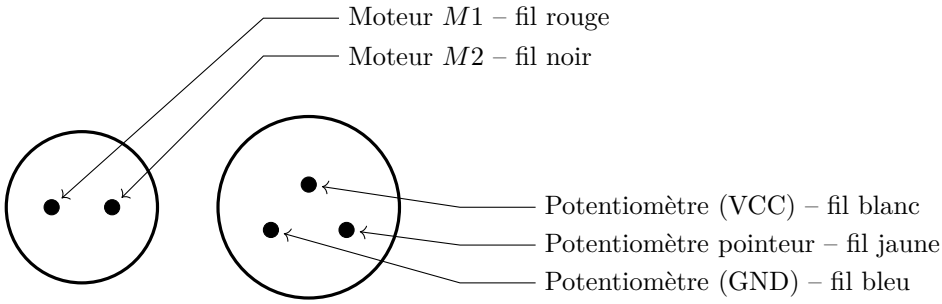
2 Commande du moteur

2.1 Commande simple

Dans cette partie on s'intéresse à mettre en place une commande simple du moteur. Vous devez réunir le matériel :

- 1 carte pyboard ;
- 1 circuit VNH 5019 ;
- 1 alimentation continue ;
- 1 plaque de prototypage (pour montage sans soudure) ;
- des fils de prototypage.

Les deux cables du vérin électrique sont respectivement :



La commande du moteur sera réalisée par l'intermédiaire d'un circuit VNH 5019, comprenant un hacheur pouvant alimenter un moteur jusqu'à 12 A et d'une alimentation continue à régler sur 12 V.

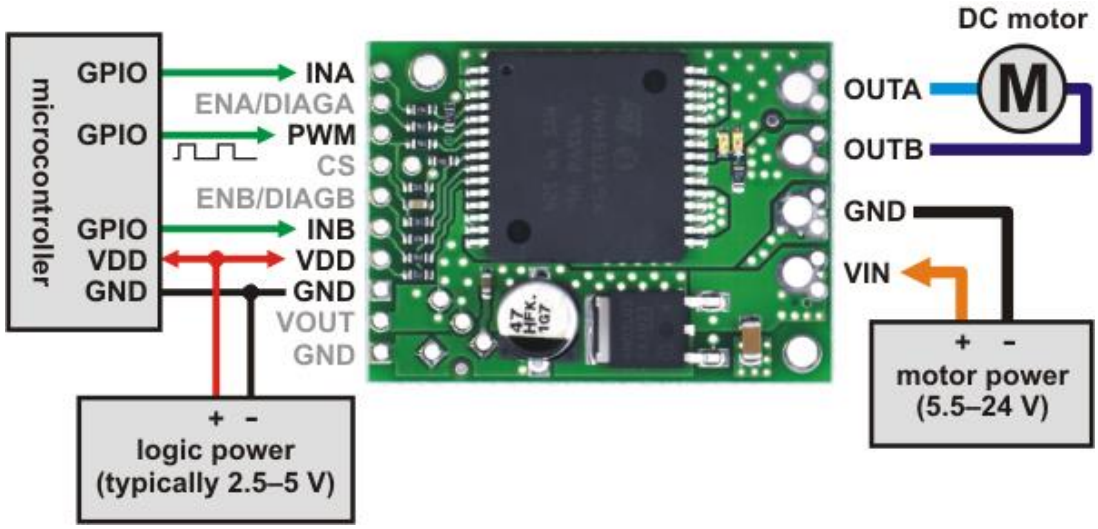


FIGURE 4 – Circuit VNH 5019.

Manipulation 2.
Coté sortie du circuit VNH 5019, connecter les voies `OUTA` et `OUTB` aux bornes du moteur (fils rouge et noir).
Connecter la voie `VIN` au 12 V et la voie `GND` au 0 V de l'alimentation du moteur.

La commande logique du hacheur VNH 5019 est réalisée entre 2,5 et 5 V, donc avec une logique directement compatible avec les 3,3 V de la carte pyboard.

Manipulation 3.
Connecter les voies du circuit VNH 5019 aux voies de la carte pyboard données ci-dessous.

VNH 5019	pyboard
PWM	X1
INA	X2
INB	X3
VDD	3.3
GND	GND

Faites vérifier votre montage par le professeur avant de mettre le système sous tension.

Initialiser ensuite une connexion série avec la carte pybord puis taper le code python suivant qui permet d'initialiser les deux sorties pour la commande du moteur.

```
from pyb import Pin   # « pour définir les entrées-sorties
from pyb import Timer # « pour le PWM
tim1 = Timer(2,freq=1000)
# *** X1 - PWM
m = Pin('X1')
ch1 = tim1.channel(1,Timer.PWM,pin=m)
# *** X2 - sens 1
s1 = Pin('X2', Pin.OUT_PP)
# *** X3 - sens 2
s2 = Pin('X3', Pin.OUT_PP)
```

Ensuite, pour commander le mouvement du vérin dans un sens faire :

```
s1.high()
s2.low()
ch1.pulse_width_percent(valeur)
```

et pour l'autre :

```
s1.low()
s2.high()
ch1.pulse_width_percent(valeur)
```

où `valeur` doit être comprise entre 0 et 100 car *width percent* signifie largeur d'impulsion en pourcentage. Enfin, pour l'arrêter, faire :

```
s1.low()
s2.low()
ch1.pulse_width_percent(0)
```

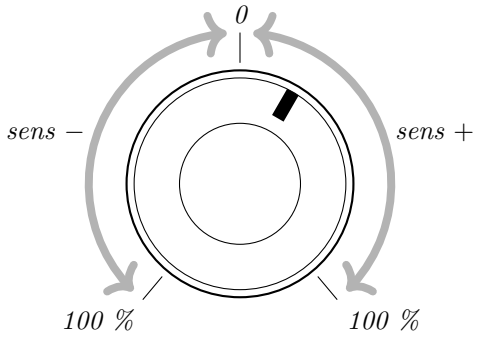
2.2 Commande simple avec un potentiomètre

Pour faciliter la commande, au lieu de taper la valeur dans le *shell*, vous allez dans cette partie mettre en place une commande avec un potentiomètre. Vous devez réunir le matériel :

- 1 carte pyboard ;
- 1 circuit VNH 5019 ;
- 1 alimentation continue ;
- 1 potentiomètre de quelques kΩ ;
- 1 plaque de prototypage (pour montage sans soudure) ;
- des fils de prototypage.

Manipulation 4.
Connecter les deux fiches extérieures du potentiomètre aux voies GND et 3,3 V de la carte pyboard.
Connecter la fiche du milieu à la voie `X5`.

Question 1. *Réaliser un programme permettant d'utiliser le potentiomètre pour moduler la vitesse de déplacement du chariot comme décrit sur la figure ci-dessous.*



où le 0 *correspond à la position médiane du potentiomètre et les deux 100 % les deux butées.*

Réaliser votre programme avec `pyzo` puis le copier vers le fichier `main.py` sur la carte pyboard. Faire un *soft reboot* avec la combinaison de touches `Ctrl+D` dans le *shell* pour redémarrer le programme principal et donc lancer le votre.

3 Asservissement de position

On s'intéresse maintenant à mettre en place un asservissement de position.

3.1 Acquisition des informations de position

Le vérin électrique est muni d'un potentiomètre rotatif fixé sur un axe entraîné en rotation par un système pignon-crémaillère par la tige mobile du vérin. Vous allez dans cette partie définir l'acquisition de la position linéaire du vérin. Vous devez réunir le matériel :

— 1 carte pyboard ;
— 1 plaque de prototypage (pour montage sans soudure) ;
— des fils de prototypage.

Manipulation 5.
Connecter les voies VCC et GND du potentiomètre du vérin aux voies 3,3 V et GND de la carte pyboard.
Connecter le pointeur du vérin à la voie `X5` de la carte pyboard.

Puis déclarer une entrée analogique avec le code suivant.

```
import pyb
pos = pyb.ADC('X5')
```

Pour lire la position linéaire du vérin, il suffit ensuite d'exécuter la commande :

```
pos.read()
```

Question 2. *Définir une fonction* `position` *permettant de renvoyer la position absolue de la tige du vérin.*

Pour vérifier votre programmation, vous imposerez différents mouvements à la tige et afficherez dans le shell la valeur de la position de la tige.

3.2 Asservissement de position

Dans cette dernière partie, vous allez mettre en place un asservissement de position en intégrant toutes les contributions élémentaires des parties 2.1 et 3.1. Vous devez réunir le matériel :

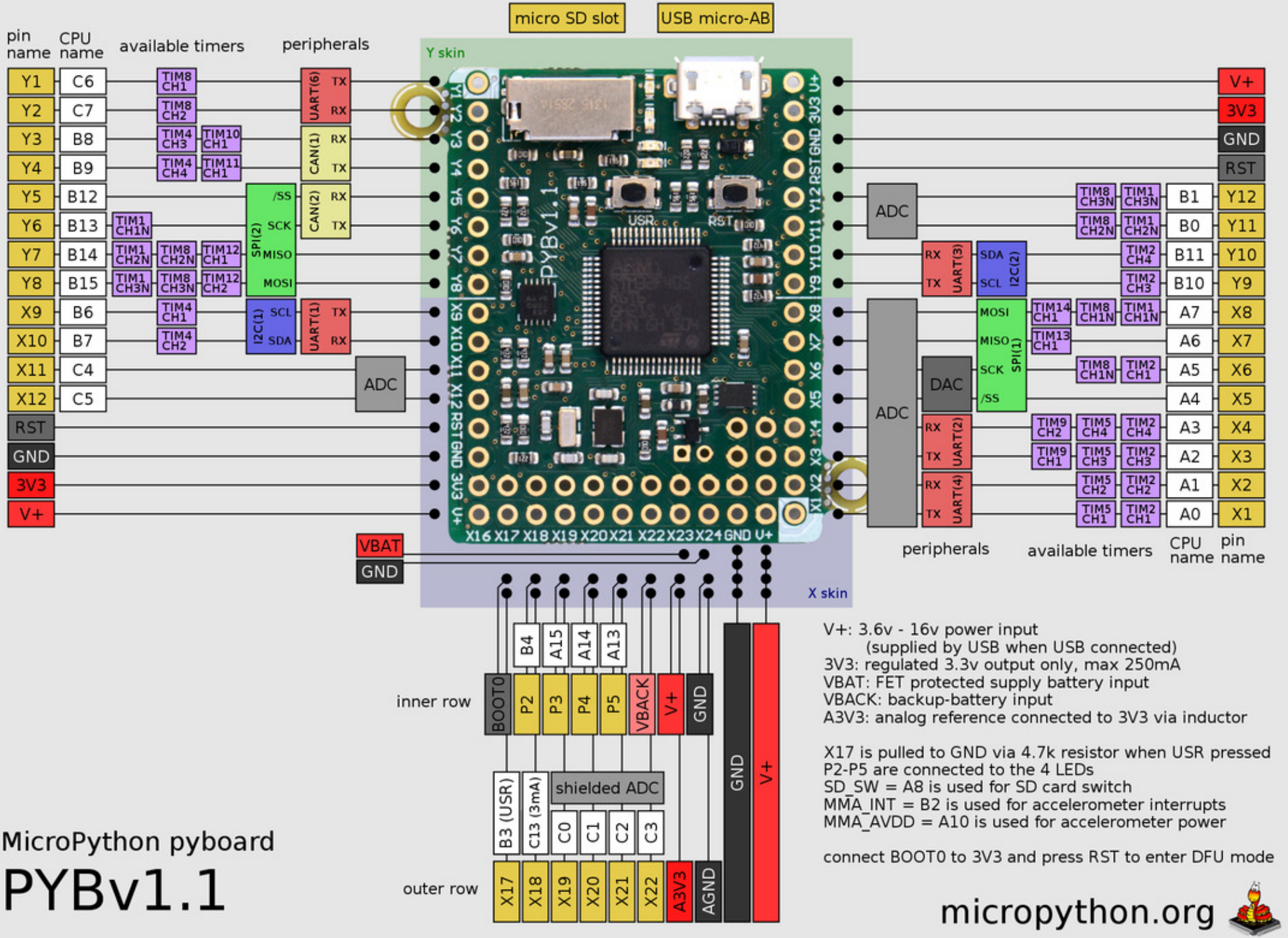
— 1 carte pyboard ;
— 1 circuit VNH 5019 ;
— 1 alimentation continue ;
— 1 plaque de prototypage (pour montage sans soudure) ;
— des fils de prototypage.

Manipulation 6.
Réaliser le câblage avec tous les éléments nécessaires et le faire vérifier.

Question 3. *Réaliser un programme permettant de déplacer la tige du vérin le plus rapidement et précisément possible à partir d'une consigne de position donnée dans le* shell.

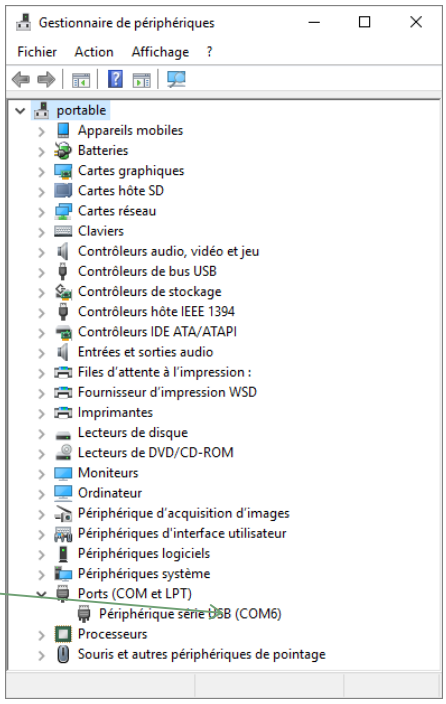
* *
*

Annexes

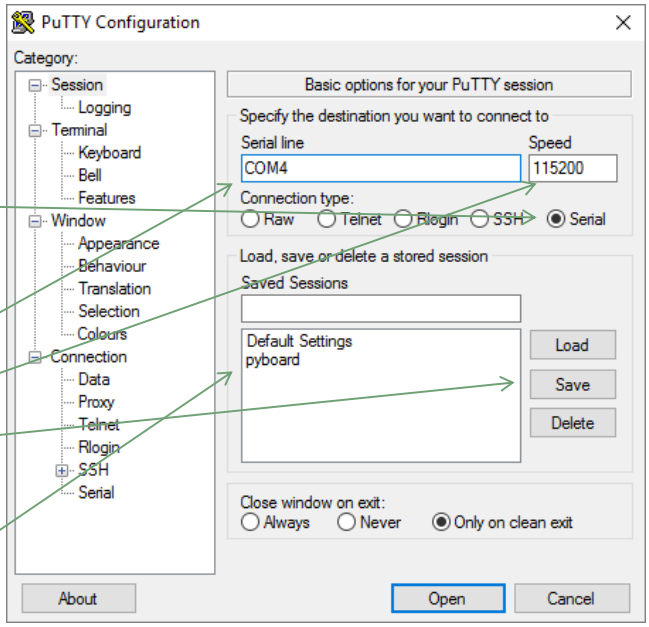

micro SD slot
USB micro-AB
pin name
CPU name
available timers
peripherals
Y skin
X skin
PYBv1.1
USR
RST
ADC
DAC
VBAT
GND
inner row
outer row
shielded ADC
peripherals
available timers
CPU name
pin name
V+: 3.6v - 16v power input
(supplied by USB when USB connected)
3V3: regulated 3.3v output only, max 250mA
VBAT: FET protected supply battery input
VBACK: backup-battery input
A3V3: analog reference connected to 3V3 via inductor
X17 is pulled to GND via 4.7k resistor when USR pressed
P2-P5 are connected to the 4 LEDs
SD_SW = A8 is used for SD card switch
MMA_INT = B2 is used for accelerometer interrupts
MMA_AVDD = A10 is used for accelerometer power
connect BOOT0 to 3V3 and press RST to enter DFU mode
MicroPython pyboard
PYBv1.1
micropython.org

Instructions connexion

- La carte pyboard est branchée à l'ordinateur par le port USB. Ce port sert aussi à son alimentation électrique.
- Le PC la voit comme une carte flash sur laquelle vous allez copier vos programmes python (fichiers *.py)
- Pour une première connexion de cette carte avec ce PC, il faut connaître le numéro du port COM auquel elle est associée : ouvrir le gestionnaire de périphériques et relever le numéro.
- Vous devrez refaire cette manipulation si vous changer de carte ou de PC.



- Ouverture du shell de la pyboard :
- Lancer le logiciel PuTTY.exe situé sous C:\
- Choisir le type de connexion : Serie
- Choisir le port (reporter le numéro du COM relevé précédemment)
- Choisir la vitesse : 115200 bauds
- Vous pouvez sauvegarder cette configuration ici.
- A la prochaine connexion vous double cliquerez sur votre config (pyboard par ex.).



Instructions shell

- Vous pouvez tester une instructions python de base
- Pour relancer le shell sans déconnecter la carte (soft reboot) appuyer sur Ctrl-D et lancer main.py.
- Pour stopper un programme qui boucle appuyer sur Ctrl-C
- Pour lancer un programme (test3.py par exemple) déjà sur la carte flash, taper
  >>> exec(open('test3.py').read())
- Il faut se déconnecter avant l'éjection de la carte en fermant la fenêtre PuTTY.
- main.py est le programme lu au lancement de la carte. Attention à son contenu !