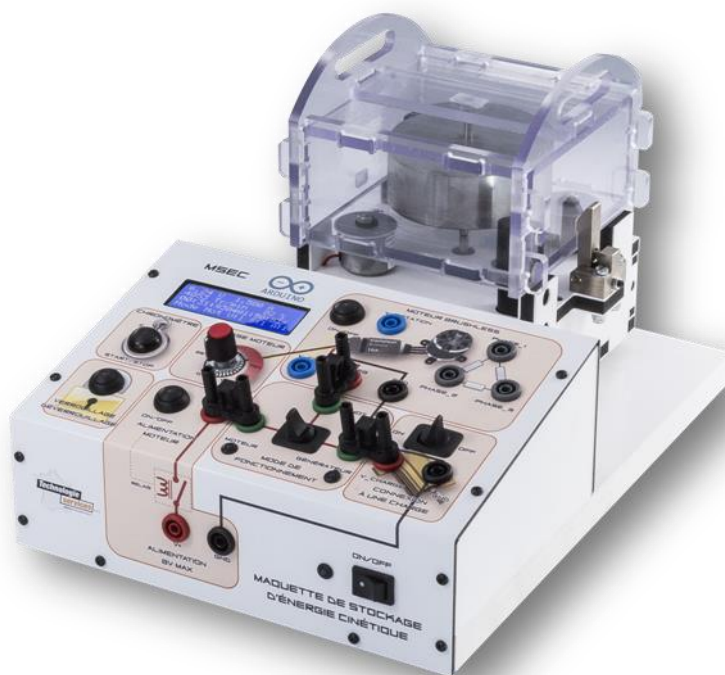




DOSSIER TECHNIQUE





# Sommaire

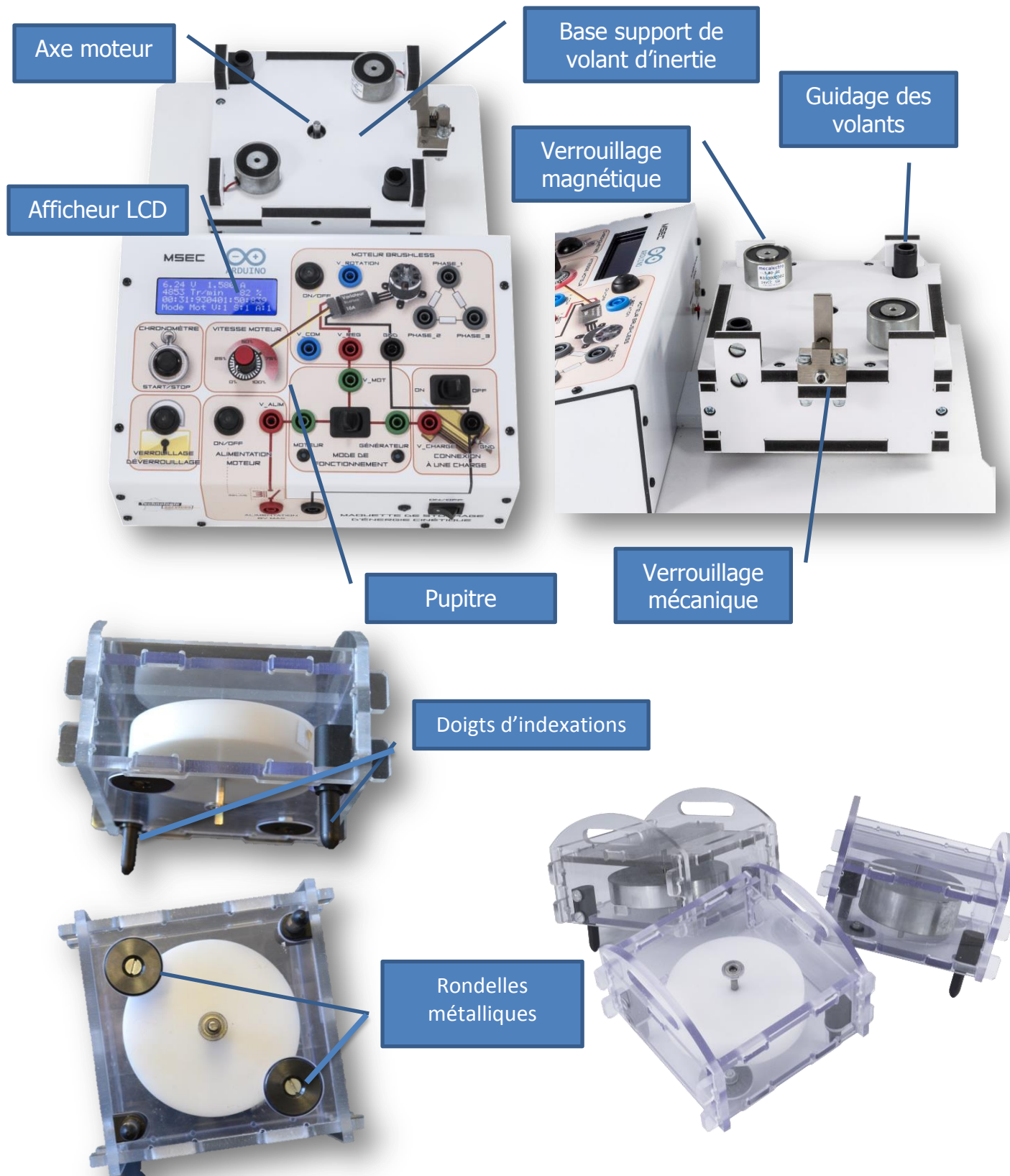
---

Descriptif rapide du produit : .....	3
Le stockage d'énergie cinétique, introduction.....	5
Le stockage d'énergie cinétique, les constituants.....	5
La platine .....	6
Tension et courant .....	7
Capteur de vitesse .....	7
Affichage des données .....	8
Schéma de câblage des capteurs .....	9
Transmission des données en USB.....	10
Le chronométrage .....	11
Le verrouillage magnétique.....	12
Le verrouillage mécanique .....	13
La charge interne.....	13
Étage de sécurité .....	13
Mode automatique .....	13
Tableau de connexion de l'Arduino.....	14
Schéma de câblage de l'Arduino .....	15
Schéma de principe de pilotage des relais .....	15
Les volants d'inertie .....	16
Interface de mesure .....	17
Face avant.....	17
Les boutons .....	17
Statut de la maquette.....	18
Diagramme .....	19
Schéma de principe de la maquette MF8 .....	20



Descriptif rapide du produit :

La platine MF8 est une maquette dédié à l'étude du stockage de l'énergie cinétique. Trois volants d'inertie sont livrés afin de pouvoir comparer les différents comportements des volants d'inertie.





## Condition pour le démarrage du moteur

Voici la liste des conditions de démarrage du moteur :

- Platine alimentée (12V) et allumée (ON),
- Volant d'inertie présent sur la platine et correctement verrouillé (verrouillage mécanique),
- Potentiomètre de « Vitesse moteur » au minimum,
- Bouton « Verrouillage » activé (ventouses),
- Bouton « Alimentation moteur » activé,
- Commutateur de « Mode de fonctionnement » sur « Moteur »,
- Cavaliers présent entre les douilles V\_ALIM/Moteur, V\_MOT/V\_REG et Générateur/V\_CHARGE
- Alimentation sur les douilles V+ et GND supérieur à 6V (maximum 8V).

Pour rappel, l'afficheur donne des informations sur l'état de la platine.

Il s'agit de l'état des capteurs de la platine afin de savoir si les conditions de démarrage moteur sont remplies.

- V (Ventouse): signifie l'état d'activation du verrouillage magnétique. Si les ventouses sont activées (volant d'inertie verrouillé) la valeur est à 1, sinon la valeur est à 0.
- S (Sécurité): signifie que le volant d'inertie est correctement placé sur sa base et que le verrouillage magnétique est activé.
- A (Alimentation): signifie que l'alimentation du moteur est validé par l'interrupteur « Alimentation moteur ON/OFF ». Ce bouton active un relais ce qui connecte la borne « V+ » à la douille « V\_ALIM » (alimentation moteur).



**Le moteur démarrera uniquement si ces conditions sont toutes remplies ( V :1, S :1, A :1).**

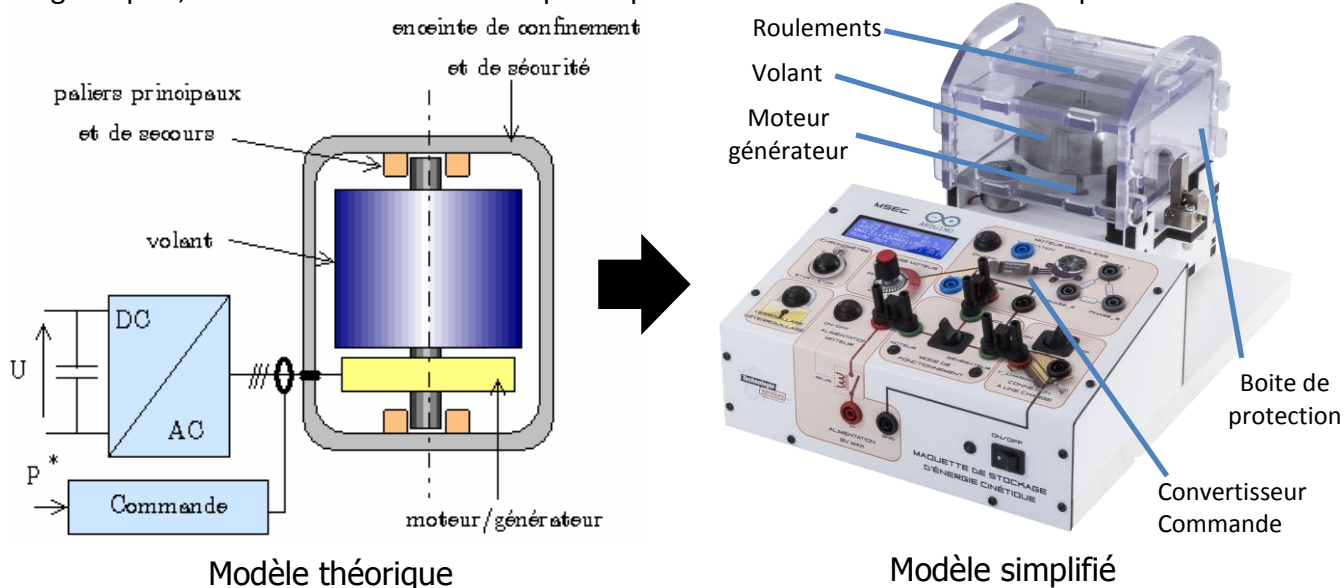


## Le stockage d'énergie cinétique, introduction

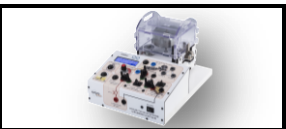
Le stockage cinétique ou inertiel est un moyen connu depuis l'antiquité (tours pour poterie). Il a ensuite été utilisé mécaniquement (transmission mécanique à rapport continûment variable) puis électriquement (Gyrobuss, Yverdon, années 1950) dans le domaine des transports pour réduire les appels de puissance ou pour éviter les caténaires (recharge rapide en station). Il a également trouvé des applications électromécaniques dans l'industrie, notamment pour les laminaires, encore pour lisser les appels de puissance électrique, mais également pour se prémunir des coupures d'alimentation électrique de faible durée. Dans la continuité de ces systèmes, les groupes électrogènes à inertie renforcée constituent des dispositifs à stockage inertiel électromécanique. Un nouvel engouement est apparu dans les années 1980 pour les automobiles électriques, car l'arrivée à maturité des matériaux composites (légers et à très haute résistance mécanique) offrait des perspectives très attractives. Les difficultés liées aux fortes contraintes mécaniques des applications embarquées (effet gyroscopique, résistance au crash, etc...) ont sévèrement réduit le champ d'applications dans le domaine des transports. Il semble aujourd'hui que les transports guidés (tramways notamment) constituent un champ d'applications réel tant en situation embarquée que stationnaire (lissage des pointe de puissance pour la source et donc moindre dimensionnement). Le second champ d'applications des accumulateurs à volant d'inertie se trouve dans les alimentations ininterrompibles où les appels de puissance de courte durée sont très sollicitant pour les batteries électrochimiques et nuisent à leur durée de vie. Notons enfin que les volants d'inertie sont utilisés dans certaines applications spatiales à la fois pour transférer de l'énergie et pour stabiliser ou orienter (effet gyroscopique) les satellites.

## Le stockage d'énergie cinétique, les constituants

Les principaux composants d'un dispositif de stockage électromécanique inertiel sont schématisés ci-dessous. On trouve ainsi en premier lieu le **volant** : élément principal de l'accumulation, puis le moteur-générateur, le tout guidé par des paliers éventuellement magnétiques, un convertisseur électronique de puissance et une enceinte à basse pression.

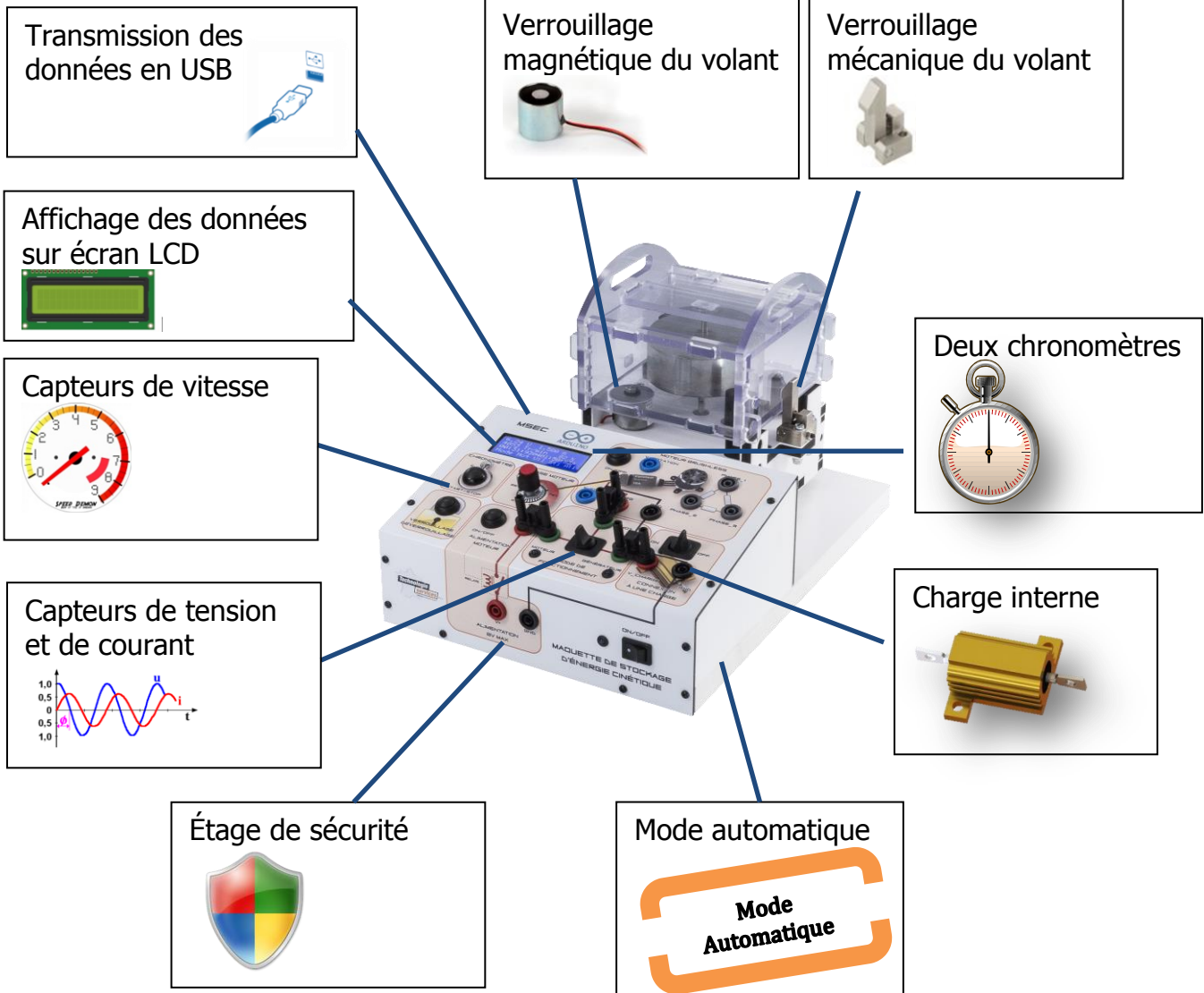


Source : Revue 3EI n°48, mars 2007, pp. 18-29



**La platine**

Afin de pouvoir étudier le système, plusieurs capteurs et fonctions ont été installés sur dans la platine





### Tension et courant

Un capteur de courant (+/-15A) et de tension (+/-30V) sont présents dans la platine, ils sont câblés comme ci-dessous derrière le variateur moteur.

Ces capteurs sont câblés sur l'Arduino qui est utilisé comme centrale d'acquisition.

Des documentations techniques des éléments sont disponibles dans le dossier ressources.



Capteur de tension



Capteur de courant

#### Mesure de courant via l'Arduino :

Le capteur de courant retourne une tension de 0 à 5V, voici le code Arduino pour obtenir le courant consommé.

```
void Mcourant() //fonction de mesure de courant
{
  {
    courantM = analogRead(incourant); // lecture de l'image du courant sur la broche incourant soit A1
    courant = (0.00488 * courantM - 2.5) / 0.11; // conditionnement du signal
    courant = courant + 0.34; // offset
  }
}
```

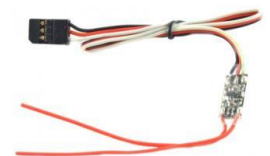
#### Mesure de tension via l'Arduino :

Le capteur de tension retourne une tension de 0 à 5V, voici le code Arduino pour obtenir la tension aux bornes du variateur.

```
void Mtension() //fonction de mesure de tension
{
  voltageInput = analogRead(intension); // lecture de l'image de la tension sur la broche intension soit A0
  voltageConvert = map(voltageInput, 0, 1023, 0, 999); // conditionnement du signal
  tension = (voltageConvert / 13.62) - 36.7107 + 0.22; // conditionnement du signal
  tension = tension + 0.5; // offset
}
```

### Capteur de vitesse

Pour mesurer la vitesse un capteur tachymétrique pour moteur brushless est câblé sur 2 phases du moteur.



Capteur tachymétrique

#### Mesure de la vitesse via l'Arduino :

Ce capteur renvoie un signal TTL proportionnel au nombre de tours par minute du moteur.

Pour faciliter la programmation nous allons utiliser une bibliothèque qui retourne la fréquence via la variable « counter.hertz » du signal connecté sur la broche D3.



```
void Avitesse() // fonction de mesure de la vitesse
{
  lcd.print((((counter.hertz())*120)/7)/2); // affichage sur le LCD et conversion de la fréquence en RPM
}
```

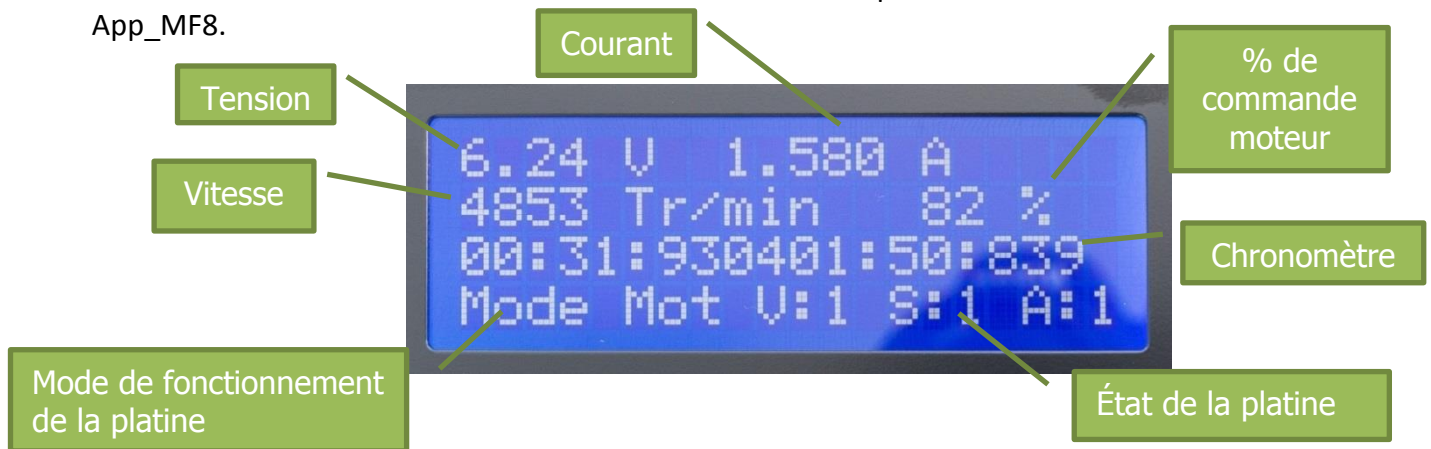
### Affichage des données

L'affichage des données s'effectue via l'Arduino connecté à un afficheur LCD 4 lignes 20 caractères. Voici les valeurs affichées sur l'écran LCD :

- Tension moteur en volts,
- Courant moteur en ampère,
- Vitesse du moteur en tours/minute,
- Commande moteur en %,
- 2 chronomètres en seconde,
- L'état de la platine.

L'affichage peut parfois scintiller, cela est dû à la vitesse de rafraîchissement qui est rapide afin privilégier l'envoyer de valeur au PC afin de pour pouvoir les exploiter au dépend d'un confort d'affichage.

Toutes ces valeurs sont transmises au PC via l'Arduino est exploitable avec l'interface App\_MF8.



### Afficher via l'Arduino :

Pour communiquer avec l'afficheur LCD nous utiliserons la bibliothèque « LiquidCrystal », cela facilitera la transmission à l'afficheur des données.

L'afficheur sera branché sur les bornes 7, 8, 9, 10, 11, 12 :

```
LiquidCrystal lcd(7, 8, 9, 10, 11, 12); // broche LCD

void Adata() // fonction affichage des données
{
  lcd.setCursor(0,0); // positionnement sur l'afficheur du curseur
  lcd.print(tension,1); // afficher la variable tension, afficher au dixième après la virgule
  lcd.print(" V "); // afficher le caractère « V »
  lcd.setCursor(8,0);
```



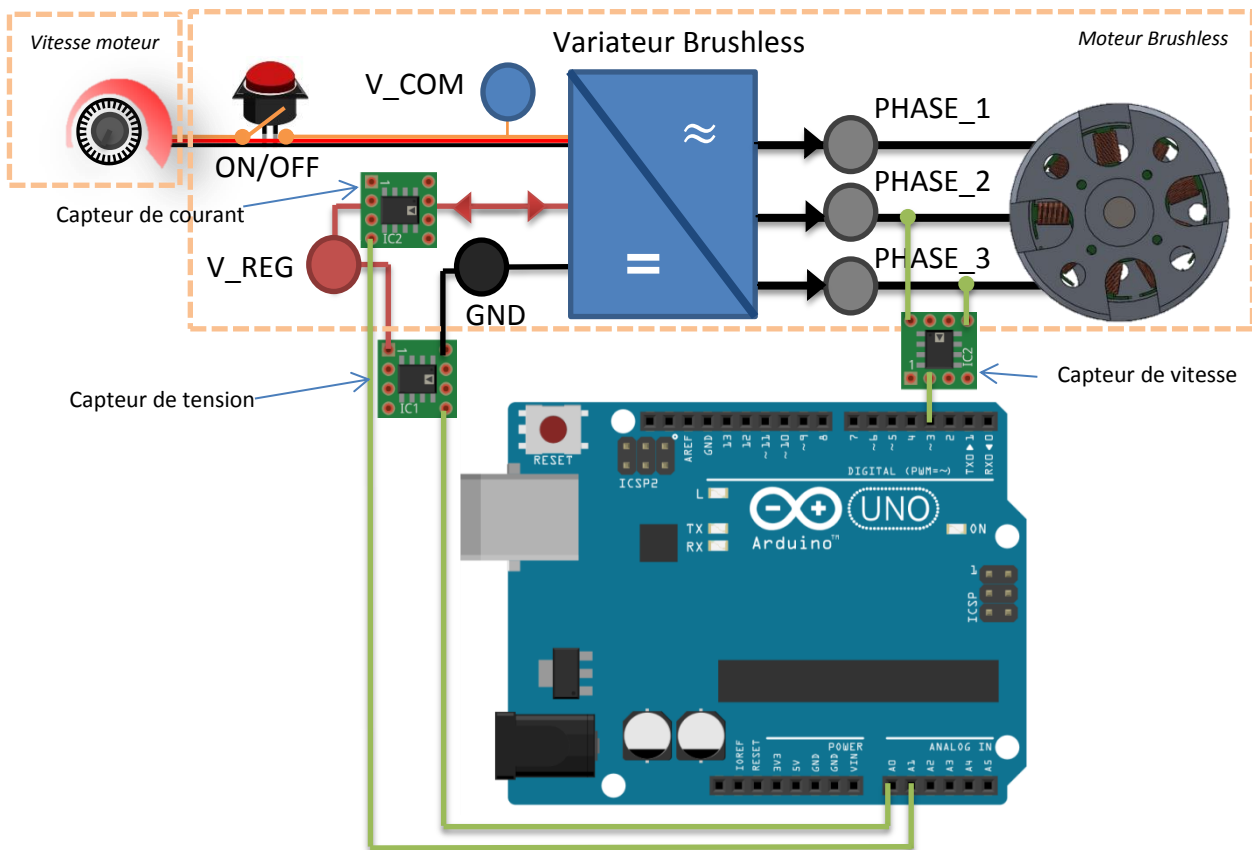


```

lcd.print(courant,1);
lcd.print(" A ");
lcd.setCursor(0,3);
if (mode ==1 ) // si la variable mode égale à 1
{ lcd.print("Mode Gen");}
Else // sinon
{ lcd.print("Mode Mot");}
lcd.setCursor(8,3);
lcd.print(" V:");
lcd.print(ventouse);
lcd.print(" S:");
lcd.print(secu);
lcd.print(" A:");
lcd.print(alim);
lcd.setCursor(15,1);
lcd.print(vitesse);
lcd.print("% ");
}

```

Schéma de câblage des capteurs



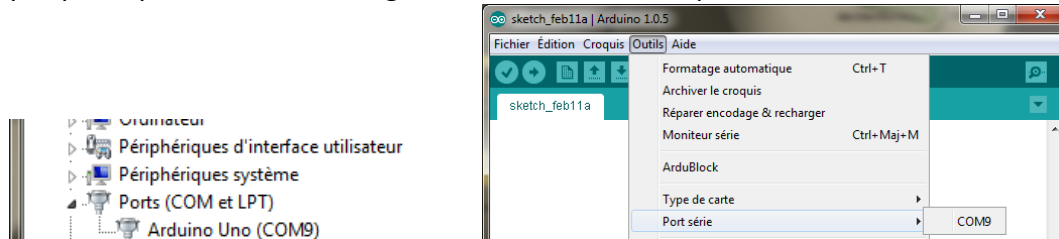
Câblage des capteurs de courant, tension et vitesse de rotation



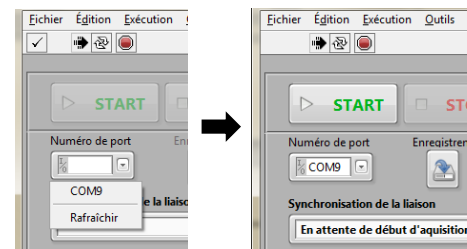
### Transmission des données en USB

La transmission des données s'effectue un port « Com » émulé.

Pour visualisé sur quel port « Com » est connecté votre Arduino, ouvrir le « Gestionnaire des périphériques » ou avec le logiciel Arduino, « Outil » puis « Port série »

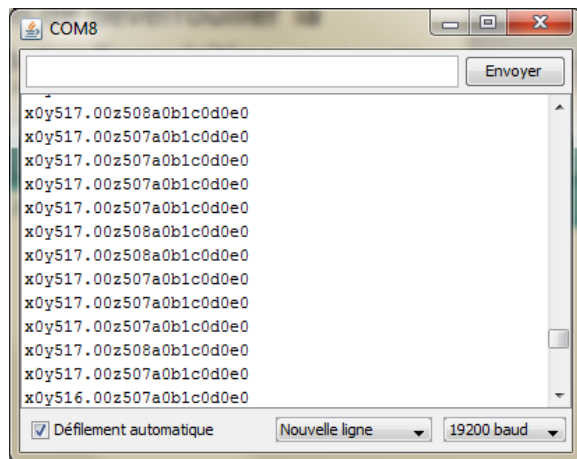


Dans l'interface AppMF8 dédiée pour la platine, il faut sélectionner le port « Com » afin de déverrouiller la fonction d'acquisition. Pour débuter l'acquisition, appuyer sur le bouton « Start ».

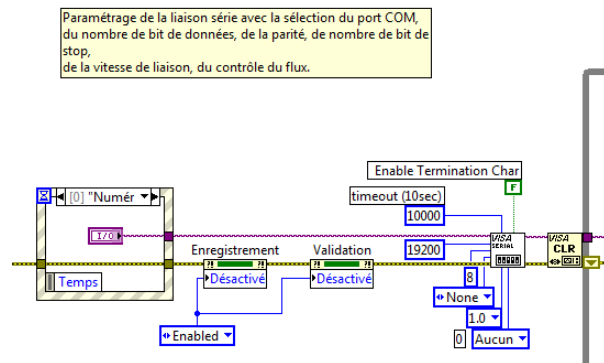


La communication avec l'Arduino s'effectue à une vitesse de 19200 bauds.

Il est possible de visualiser la trame d'information émise par l'Arduino via le « moniteur série » présent dans le logiciel Arduino.



Logiciel Arduino



Paramétrage de la liaison série de l'interface AppMF8

Programme de l'Arduino :

```
void setup(void)
{ Serial.begin(19200); //initialisation de la transmission série
```



### Le chronométrage

Deux chronomètres sont disponibles pour mesurer le temps d'accélération et de décélération du volant d'inertie. Ces chronomètres viennent en complément à l'application AppMF8.

Un seul bouton est nécessaire pour gérer le chronométrage.

Lorsque les 2 chronomètres sont à l'arrêt.

Un appui sur le bouton « Start/Stop » pour lancer le chronomètre n° 1.

Un second appui sur ce même bouton arrête le chronomètre n° 1 et lance le chronomètre n° 2.

Un troisième appui sur ce bouton arrête le chronomètre n° 2.

Pour lancer un nouveau cycle de chronométrage, appuyer à nouveau sur le bouton « Start/Stop ».



### Chronométrer via l'Arduino :

Voici le code Arduino effectuant ce chronométrage et d'affichage:

```

apres=digitalRead(startp); //variable apres prend la valeur de l'entrée pin D13 (start/stop)
if(apres==LOW&&avant==HIGH){ //Si la valeur avant =HIGH et après=LOW alors
initiation=millis(); // initialisation en milliseconde
avant=apres; //changement de valeur de la variable avant = après
bouton=0; //variable bouton = 0
while(bouton==0){ //tant que la variable bouton = à 0
Cchrono(); //calcul du chronomètre
Mcourant(); //appel sous-programme mesure de courant Mcourant
Mtension(); //appel au sous-programme mesure de tension Mtension
Mvitesse(); //appel au sous-programme mesure de vitesse Mvitesse
Statut (); //appel au sous-programme statut platine
Avitesse(); // affichage + conversion de la vitesse
Trs232(); //transmission de la chaine de caractère sur le bus série
Adata(); //affichage des données sur LCD
Achrono1(); // affiche sur LCD

apres=digitalRead(startp); //variable apres prend la valeur de l'entrée pin D13 (start/stop)
if(apres==LOW&&avant==HIGH){ //Si la valeur avant =HIGH et après=LOW alors
if(apres==LOW&&avant==HIGH){ //Si la valeur avant =HIGH et après=LOW alors
initiation=millis(); //initialisation en milliseconde
avant=apres; //changement de valeur de la variable avant = après
bouton=0; //variable bouton = 0
while(bouton==0){ //tant que la variable bouton = à 0
Cchrono(); //calcul du chronomètre
Mcourant(); //appel sous-programme mesure de courant Mcourant
Mtension(); //appel au sous-programme mesure de tension Mtension
Mvitesse(); //appel au sous-programme mesure de vitesse Mvitesse
Statut (); //appel au sous-programme statut platine
Avitesse(); // affichage + conversion de la vitesse
Trs232(); //transmission de la chaine de caractère sur le bus série
Adata(); //affichage des données sur LCD
Achrono2(); //affiche le chrono sur le LCD

apres=digitalRead(startp); //conserve la valeur de l'entrée pin D13 (start/stop)
if(apres==LOW&&avant==HIGH){ //Si la valeur avant=HIGH et après=LOW alors
avant=apres; //changement de valeur de la variable avant = après
bouton=1;} //variable bouton = 1
avant=apres; //changement de valeur de la variable avant = après
}
}

```



```

}
}
avant=apres;           //changement de valeur de la variable avant = après
}
}
avant=apres;           //changement de valeur de la variable avant = après
void Achronoz2()
{
  lcd.setCursor(9,2);
  lcd.print(" ");
  lcd.print(md);       //écrit les différentes variables du chronomètre sur l'afficheur
  lcd.print(mu);
  lcd.print(":");
  lcd.print(sd);
  lcd.print(su);
  lcd.print(":");
  lcd.print(lc);
  lcd.print(ld);
  lcd.print(lu);
}

void Cchrono()
{
  temps=millis()-initiation; //calcul du temps depuis l'activation du chronomètre
  m=temps/60000;             //calcul des minutes
  mu=m%10;                   //découpage des minutes pour obtenir l'unité
  md=(m-mu)/10;              //découpage des minutes pour obtenir les dizaines
  s=(temps/1000)-(m*60);     //Calcul des seconds
  su=s%10;                   //découpage des minutes pour obtenir l'unité
  sd=(s-su)/10;              //découpage au deuxième et prise de la valeur des dizaines
  l=temps-(s*1000)-(m*60000); //calculé millièmes de seconde
  lu=l%10;                   //découpage des seconds pour obtenir l'unité
  ld=((l-lu)/10)%10;         //découpage des seconds pour obtenir les dizaines
  lc=(l-(ld*10)-lu)/100;    //découpage des seconds pour obtenir les centièmes
}

```

### Le verrouillage magnétique

Le maintien du volant d'inertie s'effectue par un verrouillage magnétique. Il est piloté par un bouton poussoir nommé « Verrouillage/déverrouillage ».



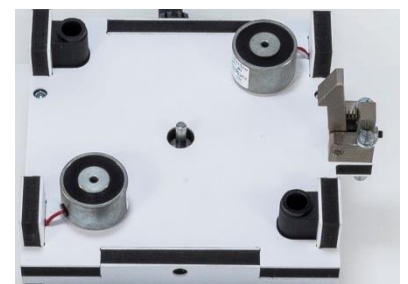
Chaque ventouse peut résister à une force de traction maximum de 27 daN (sous 24V). Les caractéristiques techniques sont présentes dans le « Dossier ressources\Ventouses ».

Dans notre cas les ventouses sont alimentées en 12V et la pièce servant de support (les rondelles) ont une épaisseur de 3mm, soit  $25\text{daN}/2 = 12.5\text{daN}$  (par ventouse) donc 25daN de force de maintien pour la platine.

Cela est suffisant pour garantir un maintien des volants d'inertie lors des expérimentations.

Les ventouses sont alimentées par la platine par le bloc secteur 220V/12V.

Les ventouses sont pilotées par un relais.



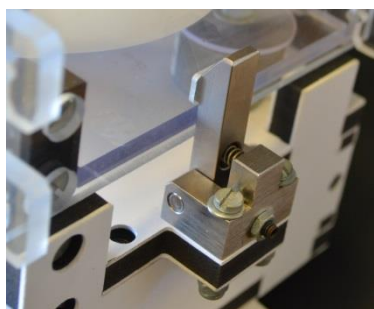


#### Le verrouillage mécanique

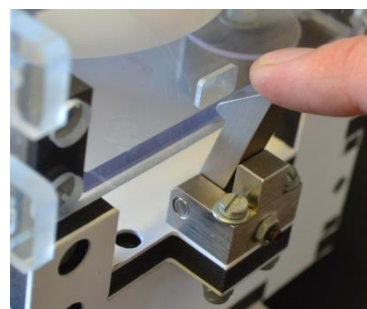
Un verrouillage mécanique est présent afin de maintenir en place le volant d'inertie lors des expérimentations en cas de coupure d'alimentation (ventouse non aimanté).

Même en cas de coupure d'alimentation de la platine, le volant en rotation restera en place et sera maintenu.

Procédure de déverrouillage :



Verrouillé



Déverrouillé

Pour info : Le moteur ne démarrera pas si les ventouses ne sont pas activées.

#### La charge interne

Dans la platine est présent une charge de 1ohm - 50W.

Il est possible de faire fonctionner le moteur à vide, de connecter la charge interne ou de connecter une charge externe (choisi par l'utilisateur).

La connexion à une charge changera le comptement de restitution d'énergie.



#### Étage de sécurité

Un étage de sécurité est présent pour garantir une protection lors de l'utilisation de la platine. Un schéma nommé « **schéma de principe de câblage des relais** » décrit cette fonction.

Des interrupteurs fin de course détectent la présence d'un volant sur sa base. L'activation du verrouillage magnétique couplé à cette détection de présence des volants activent un relais dit de « sécurité ».

#### Mode automatique

Il y a 2 possibilités de passer en « mode générateur » lorsque le volant est en rotation.

- Activer manuellement le commutateur « Mode de fonctionnement »,
- Alimenter le moteur avec une tension inférieure à 6V.

Un comparateur compare la tension de la borne « V+ » à une tension de référence de 6V.



Si la tension d'alimentation est inférieure à celle de  $V_{réf}$ , le mode « générateur » est activé automatiquement.

### Tableau de connexion de l'Arduino

Connecteur Arduino	Entrée du DAQ	Signal	type
A0	Tension moteur	0-5V	Analogique
A1	Courant moteur	0-5V	Analogique
A2	Information de la commande vitesse	0-5V	Analogique
A3	N/A	N/A	N/A
A4	N/A	N/A	N/A
A5	N/A	N/A	N/A
D0	Transmission série/USB	TTL	FTDI
D1	Transmission série/USB	TTL	FTDI
D2	État verrouillage ventouse	TTL	entrée TOR
D3	Capteur de vitesse	TTL	entrée TOR
D4	Relais mode	TTL	entrée TOR
D5	Relais sécurité	TTL	entrée TOR
D6	Relais alimentation	TTL	entrée TOR
D7	LCD	TTL	sortie
D8	LCD	TTL	sortie
D9	LCD	TTL	sortie
D10	LCD	TTL	sortie
D11	LCD	TTL	sortie
D12	LCD	TTL	sortie
D13	Pilotage des Chronomètres	TTL	entrée TOR

Ce tableau peut vous être utile si vous souhaitez réaliser des modifications de l'application fournie.

Les informations lus par le LabVIEW sont les suivantes :

Trame envoyé par l'Arduino à l'AppMF8: **x0y527.00z508a0b1c0d0e0**

**x** : vitesse moteur (information TOR, ON/OFF à une fréquence de 0-1200 Hz ,

**y** : tension moteur (information analogique, 0-1024),

**z** : courant moteur (information analogique, 0-1024),

**a** : information commande moteur (information analogique, 0-5V),

**b** : information mode moteur ou générateur (information TOR, ON/OFF),

**c** : information ventouse (information TOR, ON/OFF),

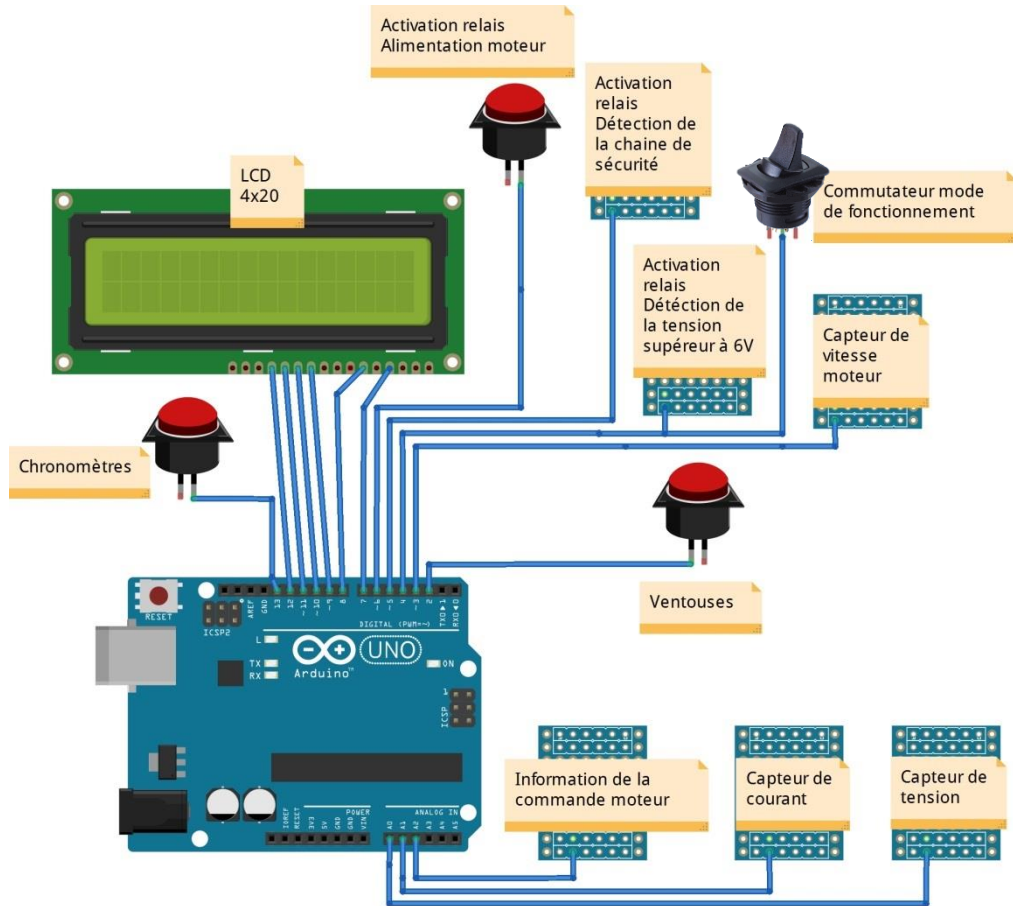
**d** : information alimentation supérieur à 6V (information TOR, ON/OFF),

**e** : information chaine de sécurité (information TOR, ON/OFF),

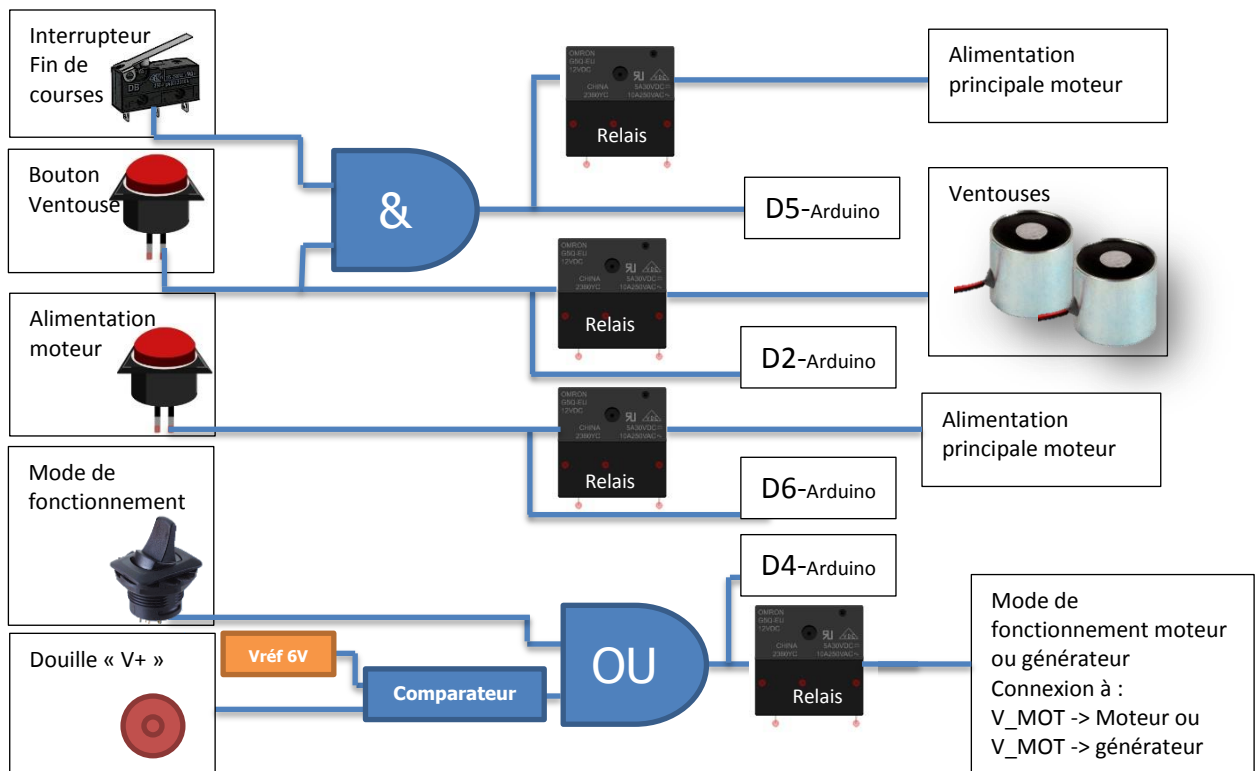
TOR : Tout Ou Rien



**Schéma de câblage de l'Arduino**



**Schéma de principe de pilotage des relais**



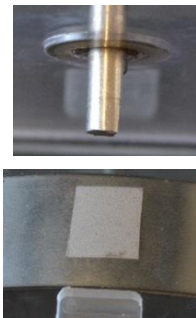


### Les volants d'inertie

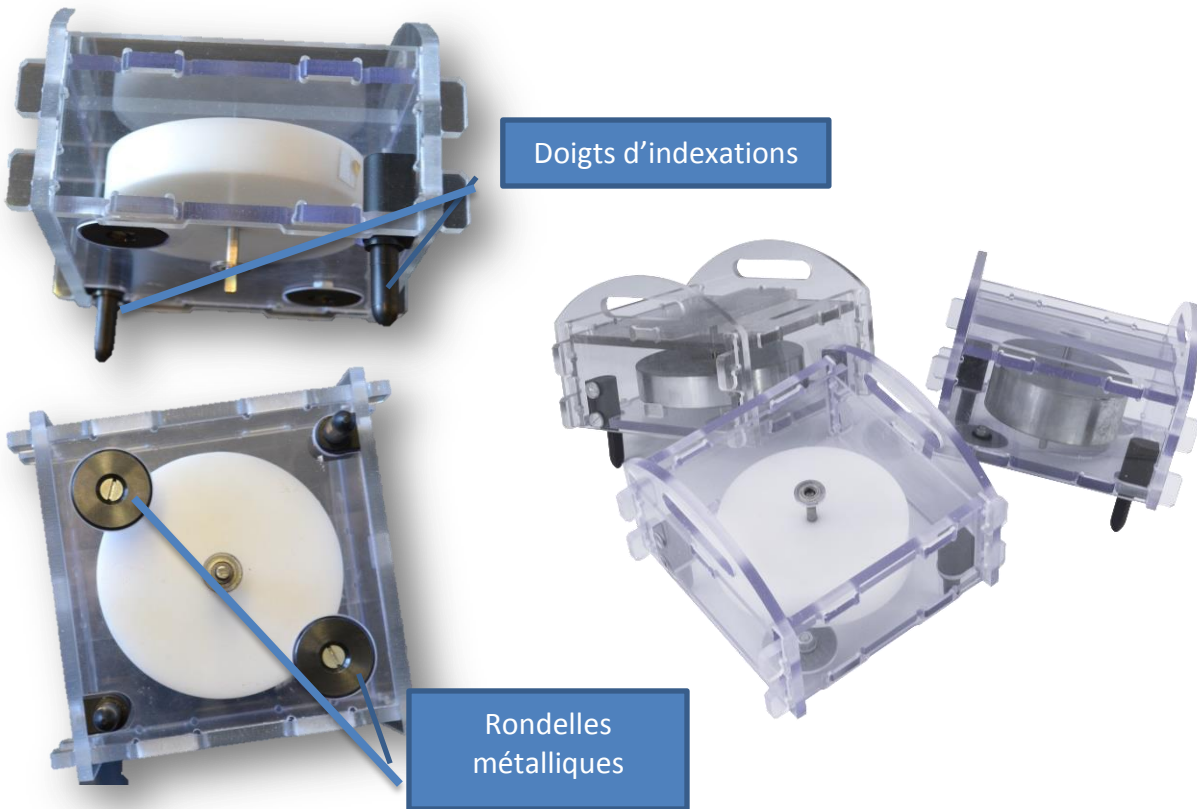
Trois volants ont été placés dans des boites en polycarbonate de 6mm afin de protéger les utilisateurs et de faciliter leurs positionnements sur la platine.

Des doigts d'indexations facilitent le positionnement des volants sur la platine.  
 Des rondelles métalliques sont également présentes sur les boites afin de permettre un verrouillage par l'intermédiaire de ventouses magnétiques.  
 Cela garantit un maintien optimum lors des expérimentations.  
 L'accouplement entre le moteur et les volants se réalise à l'aide d'une durite silicone.

L'axe est fraisé afin de permettre à la durite servant à l'accouplement de s'extraire facilement.  
 Une bande réfléchissante est présente afin de permettre la mesure de la vitesse de rotation via un tachymètre optique.  
 Chaque volant possède ce réflecteur.  
 Les plans de ces volants sont disponibles dans le « Dossier ressources\Volants d'inerties »



Volant	Matière	Dimensions en mm	Masse en kg
Volant 1	Acier zingué	30 x 80	1.2
Volant 2	PEHD	20 x 100	0.15
Volant 3	Acier zingué	20 x 100	1.2



**Les boites des volants d'inertie ne sont en aucun cas démontables. Ne jamais sortir les volants des boites sous peine de perdre le fonctionnement de la maquette et de perte de garantie**





### Interface de mesure

Toutes les mesures faites sur la platine MF8 sont regroupées sur une interface LABVIEW. Un programme LABVIEW est appelé "instrument virtuel" (VI).

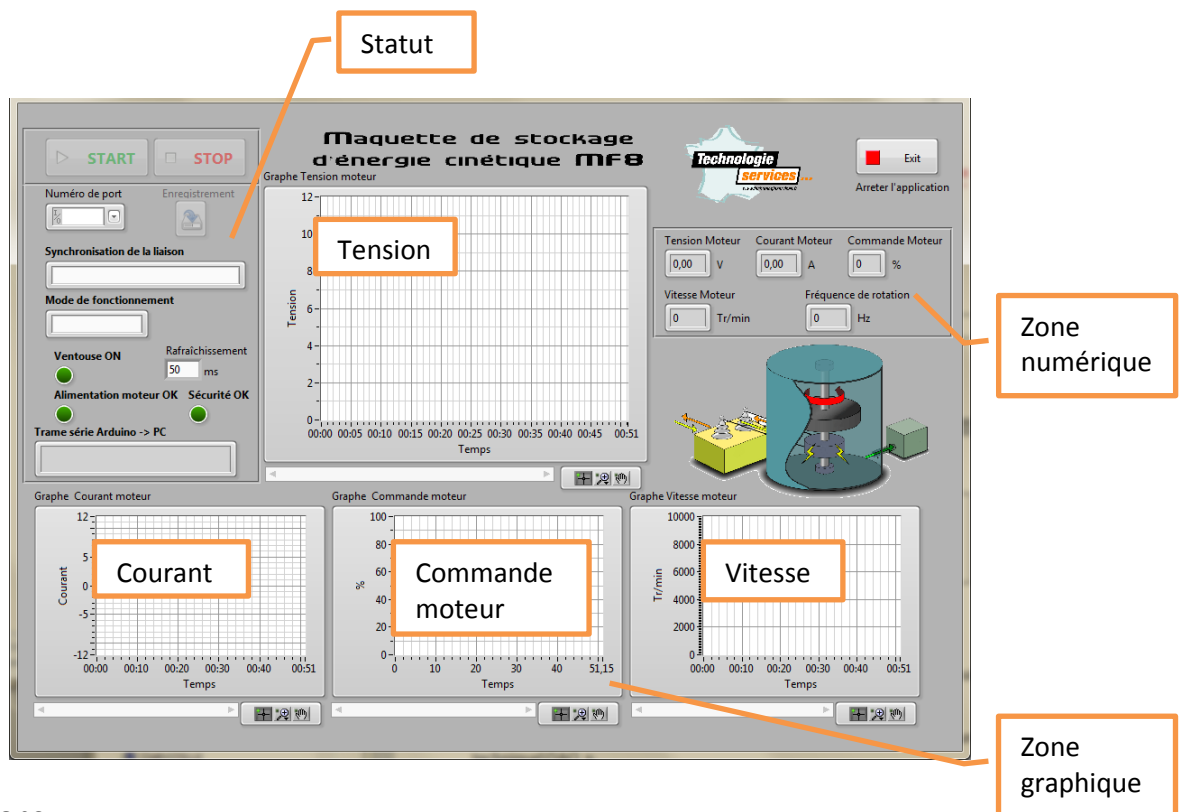
La programmation de cette interface est faite à l'aide de deux fenêtres liées. La première est la **face avant** qui contient les contrôles et les afficheurs de données (Interrupteurs, potentiomètres avec curseurs, boîtes de dialogue, LED...).

La deuxième est le **diagramme** qui contient le programme (terminaux de commandes, constantes, terminaux d'affichage, des connecteurs, des fils...).

Un document complet sur l'utilisation de l'interface Labview est disponible dans le Dossier ressources\Application Labview.

### Face avant

L'interface est constituée de zone de graphique, d'une zone de statut de la platine pour les différentes mesures électriques de la platine.

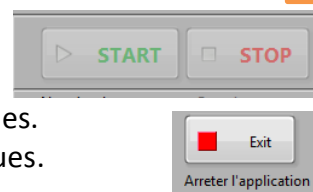


### Les boutons

START : démarrage de l'acquisition.

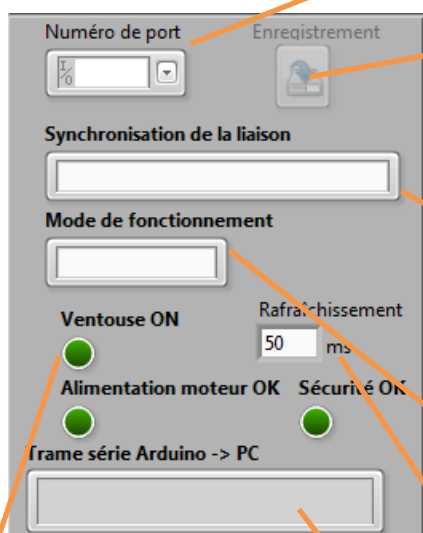
STOP : Arrêt de l'acquisition. Permet l'exploitation des graphiques.

EXIT : Arrêt la connexion USB avec l'Arduino, efface les graphiques.





### Statut de la maquette



Sélection du port de communication avec l'Arduino

Enregistrer les valeurs affichées au format tableur

Indique si la communication avec la maquette est établie :

En attente de début d'acquisition ...

OU

Synchronisée et fonctionnelle

Indique si la maquette est en mode « Générateur » ou « Moteur ».

Rafraichissement des données toutes les 50ms. (valeur par défaut)

Caractère visible transmis par l'Arduino

État des fonctions de la maquette:

- Ventouses (verrouillage par ventouse activé ou non).
- Alimentation moteur (relais d'alimentation moteur activité ou non).
- Sécurité (Condition de démarrage moteur valide ou non).

Pour information :

Les conditions de démarrage du moteur sont les suivantes :

1. Platine alimentée (12V) et allumée (ON),
2. Volant d'inertie présent sur la platine et correctement verrouillé (verrouillage mécanique),
3. Potentiomètre de « Vitesse moteur » au minimum,
4. Bouton « Verrouillage » activé (ventouses),
5. Bouton « Alimentation moteur » activé,
6. Commutateur de « Mode de fonctionnement » sur « Moteur »,
7. Cavalier présent entre les douilles V\_ALIM/Moteur, V\_MOT/V\_REG et Générateur/V\_CHARGE
8. Alimentation sur les douilles V+ et GND supérieur à 6V (maximum 8V).

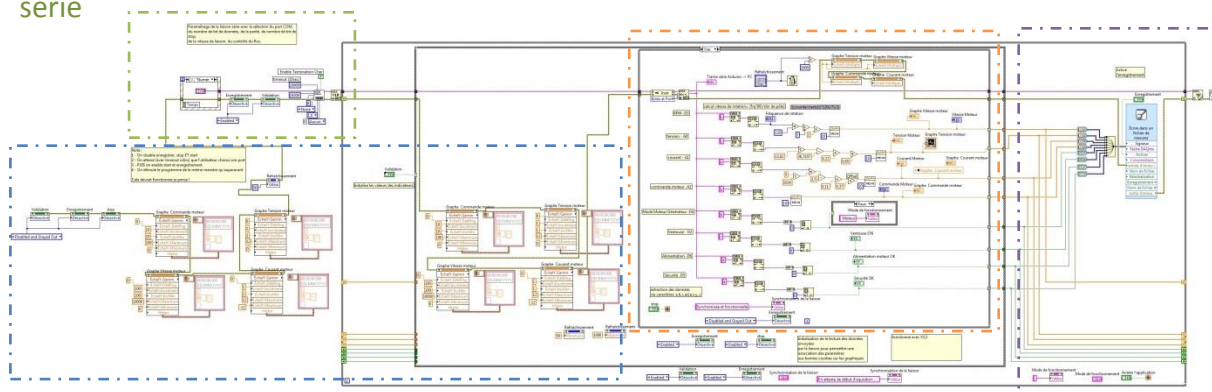


Diagramme

Le diagramme contient le code graphique du programme Labview (VI). La programmation est graphique selon une logique de flux de données. Les commandes et indicateurs de la face avant figurent sous forme de terminaux dans la fenêtre diagramme et sont à connecter entre eux selon le traitement désiré.

Paramétrage de la liaison série

Mise en forme des mesures de la tension, du courant, de la vitesse, de l'état de la platine

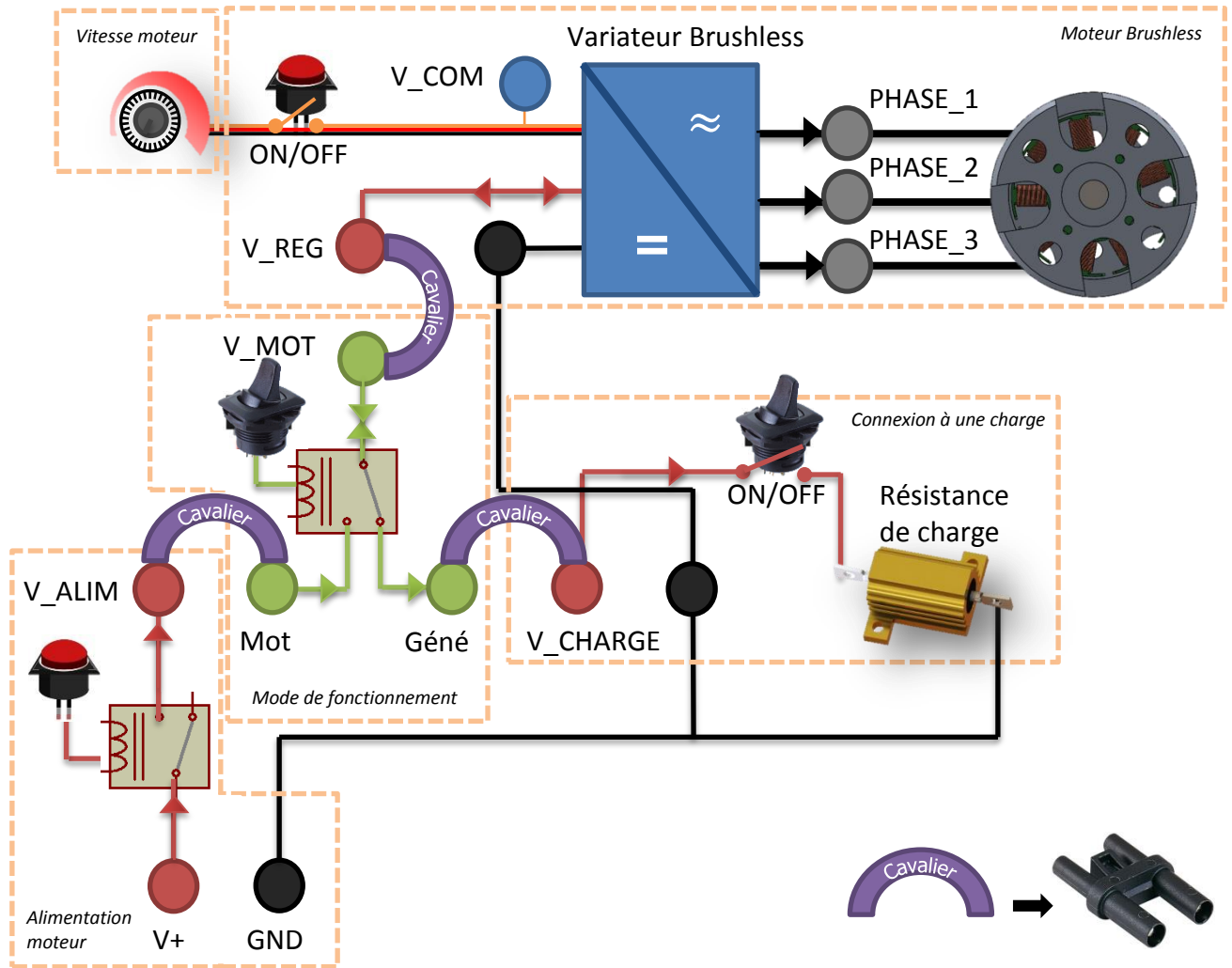


Initialisation l'interface et mise à l'échelle des zones graphiques

Enregistrement des données sur un fichier



**Schéma de principe de la maquette MF8**



La platine nécessite des cavaliers pour son fonctionnement car le circuit est ouvert afin d'utiliser des appareils de mesures conventionnelles pour les mesures électriques.