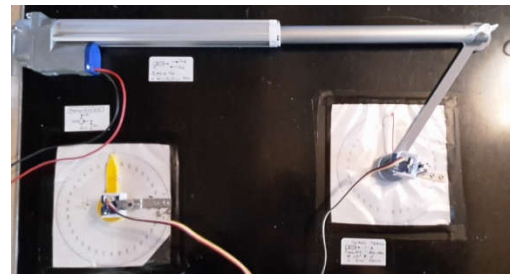
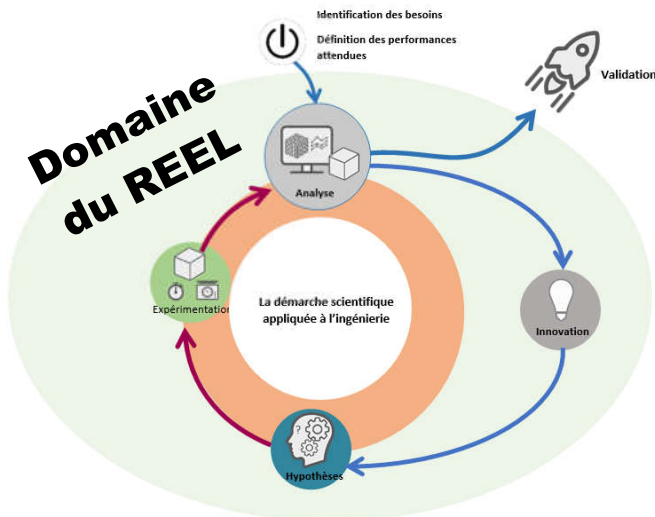


Analyse de la trame I²C « DIRECTION »



Maquette expérimentale de Pilote automatique



Objectif de cette activité :
Appréhender le contenu et les caractéristiques de cette trame.



Problème technique de l'activité 3A :

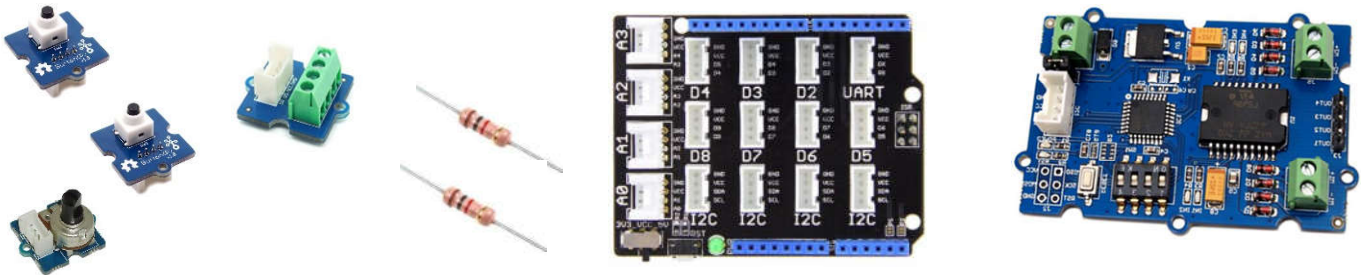
Comment et quelles informations sont communiquées lors de l'envoi de la trame « DIRECTION » ?
Le fait d'utiliser une liaison série peut-il affecter le temps de réaction du vérin ?



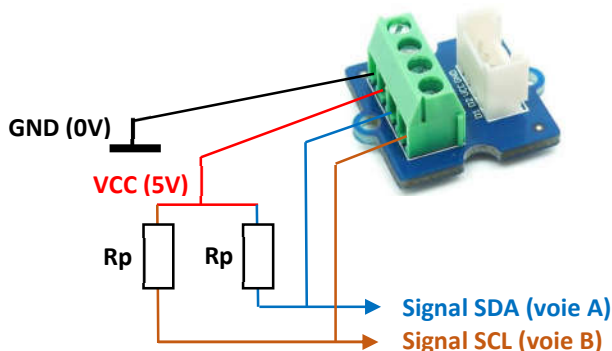
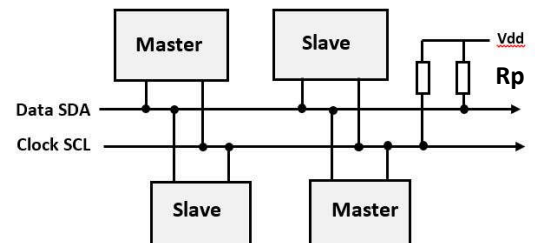
PREPARATION DE L'EXPERIMENTATION

Le matériel requis pour l'expérimentation qui suit est le suivant :

2 modules Grove Bouton Poussoir	1 carte ARDUINO UNO	1 cordon USB (PC <-> UNO)
1 module Grove potentiomètre	1 shield GROVE	1 alimentation 12 VDC
1 module Grove bornier à vis	1 Module DRIVER MCC I ² C	2 sondes d'oscilloscope (signaux SDA + SCL)
2 résistances 1 kΩ	Le moteur est optionnel donc pas obligatoire !	1 oscilloscope capable de lire les trames I ² C (exemple : PICOSCOPE)



Remarque importante : Pour observer correctement la trame I²C au PICOSCOPE, il convient d'ajouter 2 résistances Rp dites de tirage (Pull up) de valeur 1 kΩ. (valeurs variables selon la vitesse du bus I²C).



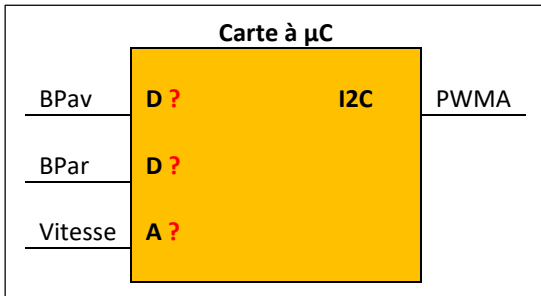


EXPERIMENTATION

Le programme de test qui suit a pour but d'aider à l'analyse et la compréhension des informations contenues dans la transmission de la trames sérielles via un BUS I²C. Rappel : la trame avec l'information vitesse (vitesse de rotation du moteur) est envoyée avant la trame d'information de direction (rentrée ou sortie tige).

Si l'on sait que le bouton poussoir BPav est rattaché à l'ordre « Sortie ou avancer la tige », BPar à l'ordre « Rentrer ou reculer la tige » et Le potentiomètre à la consigne de Vitesse :

- 1- **Déduire** du programme les broches d'affectation des entrées de la carte à microcontrôleur.



- 2- **Compléter** le pseudo-code de la boucle infinie du programme.

DEBUT
Appeler librairie <Wire.h>
Appeler librairie <MotorI2C.h>
Déclarer variable Vitesse
...

BOUCLE A L'INFINI
Lire ...
SI ... = ... ALORS
Ecrire valeur Vitesse dans PWM moteur A
Ecrire avancer dans le registre de Direction du moteur A
SI ... = ... ALORS
Ecrire valeur Vitesse dans PWM moteur A
Ecrire reculer dans le registre de Direction du moteur A
SI ... ALORS
Ecrire valeur 0 dans PWM moteur A
Ecrire valeur Vitesse dans le moniteur série
FIN SI

FIN

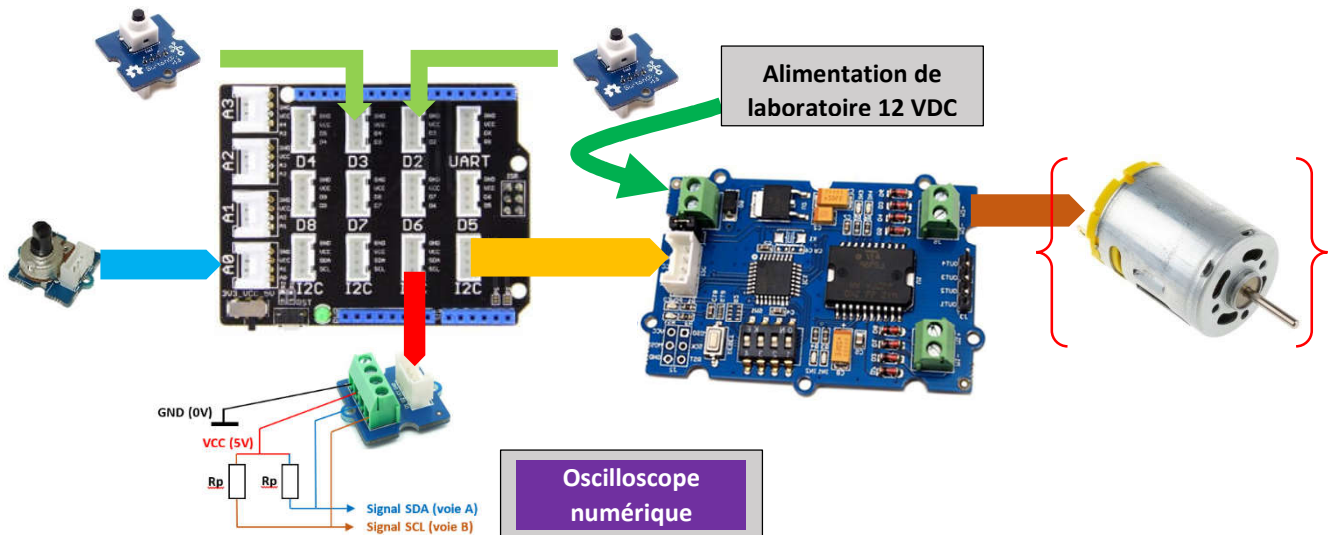
```
#include <Wire.h>
#include <MotorI2C.h>

int Vitesse = 0 ;
DigitalRead(int pinNumber)
{
  pinMode(pinNumber, INPUT);
  return digitalRead(pinNumber);
}
MotorI2C mesMoteurs;

void setup() {
  mesMoteurs.brancher();
  Serial.begin(9600);
}

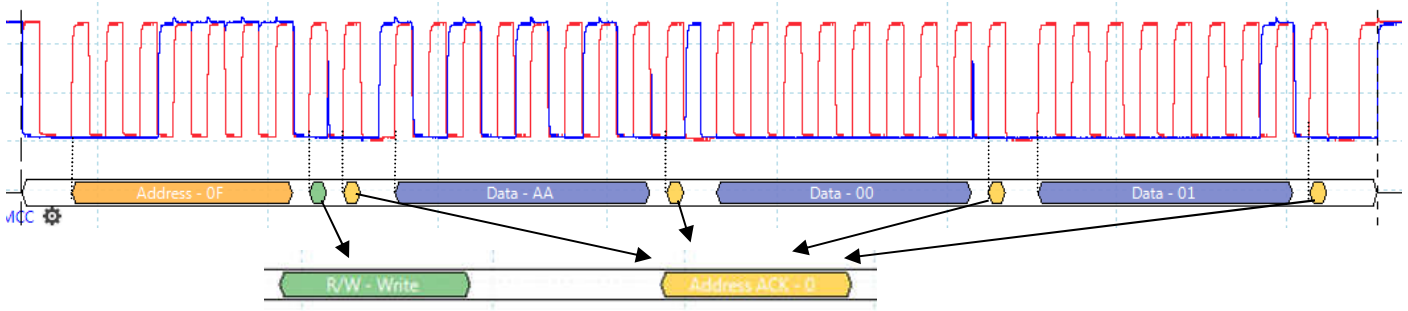
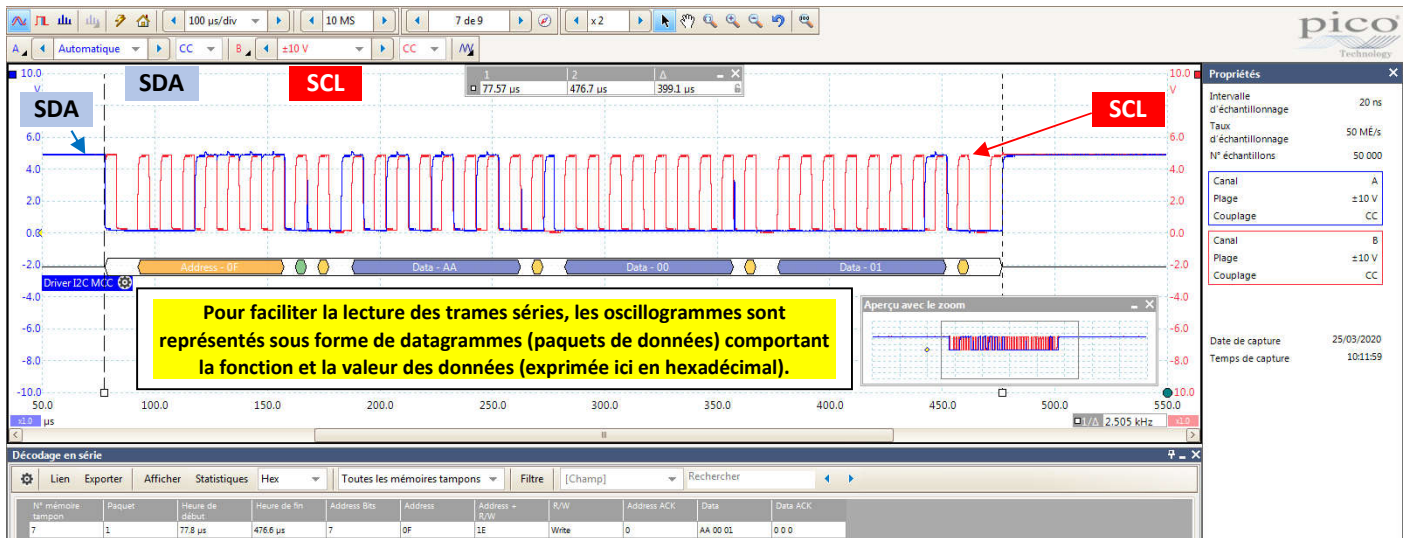
void loop() {
  Vitesse = analogRead(A1) ;
  if (DigitalRead(2))
  {
    mesMoteurs.ecrireVitesseMoteurA(Vitesse);
    mesMoteurs.avancer("tout_droit");
  }
  if (DigitalRead(3))
  {
    mesMoteurs.ecrireVitesseMoteurA(Vitesse);
    mesMoteurs.reculer("tout_droit");
  }
  if ( (!(DigitalRead(2) ) && !( DigitalRead(3) ) ) )
  {
    mesMoteurs.stopper();
  }
  Serial.print(Vitesse);
  Serial.print(" ");
  Serial.println();
}
```

- 3- Réaliser le montage permettant de visualiser les signaux SDA et SCL de la trame I2C.



- 4- Téléverser le programme dans la carte à μC via le port USB. (TestI2C.ino)
- 5- En présence du Professeur, régler le 12 VDC, régler le potentiomètre Vitesse à la valeur numérique 500 (vérifier à l'aide du moniteur série) puis mettre sous tension tout en s'assurant du bon fonctionnement du montage. Penser, si besoin, à faire un RESET sur la carte à μC ou sur le DRIVER I²C en cas de problème.
- 6- Régler l'oscilloscope et observer les trames I²C.

Après quelques réglages... Voici ce que l'on peut observer au PICOSCOPE lorsqu'aucun BP n'est actionné (Moteur à l'arrêt).



EXPLOITATION DES MESURES - Caractérisation des échanges I2C entre la carte à μC et le Driver MCC

7- Pour correctement décoder les trames I2C, consulter le document ressource sur l'essentiel de la théorie du PROTOCOLE I²C.

8- Identifier les bits et les paquets de la trame à l'écran de l'oscilloscope puis reporter le nombre de bits sur la ligne d'en-dessous.



De l'esclave vers le maître

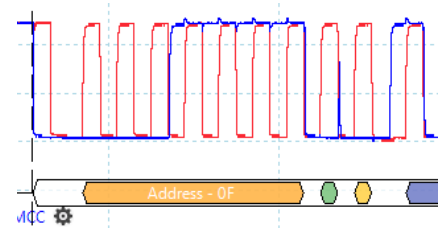
S	Adresse esclave	W	A	DATA 1	A	DATA 2	A	DATA 3	A	P
S	?	?	?	?	?	?	?	?	?	P

- 9- Les données véhiculées sont écrites dans quel composant : la carte à μC ou le Driver I²C ?
- 10- Préciser à quel moment la carte à μC est destinatrice ou expéditrice ?
- 11- L'adresse 0F_h (encore noté 0x0F) permet d'identifier quel composant : la carte à μC ou le Driver I2C ?

- 12- **Transcoder** l'adresse 0x0F en binaire (noté 0b....) grâce au chronogramme ci-contre.

La trame série d'adresse commence-t-elle par le bit de poids fort (MSB) ou le bit de poids faible (LSB) ?

(Cette réponse vaudra pour tous les autres datagrammes)



- 13- Déterminer la fréquence d'horloge f_H de la trame et la durée Δt du message.
 14- Déterminer la fréquence d'envoi f_{envoi} des messages ?
 15- La vitesse de bus I2C est-elle Standard, Rapide, Rapide Plus ou à Haute Vitesse ? Justifier votre réponse.

DECODAGE DE LA TRAME « DIRECTION »

Que signifie ce message ?

Selon que l'on appuie ou pas sur BPav ou BPar, les trames suivantes apparaissent périodiquement :

N°	Trame « DIRECTION » observée
1	Address - 0F Data - AA Data - 00 Data - 01
2	Address - 0F Data - AA Data - 05 Data - 01
3	Address - 0F Data - AA Data - 0A Data - 01

En s'aidant de l'extrait du document constructeur – DRIVER I2C proposé lors lancement de l'activité 3 :

- 16- **Préciser** ce que signifie de datagramme DATA – AA ?
 17- **Préciser** pour quel numéro de trame la tige rentre, sort ou reste à l'arrêt ?

NB : Le 4^{ème} paquet est obligatoire par respect pour le format du message (même s'il signifie « Ne rien faire » !).



CONCLUSION partielle sur le choix du DRIVER MOTEUR I²C

- 18- En comparaison avec la solution à technologie TOR à relais imaginée en première intention, cette solution de commande faisant appel à de la transmission série au format I²C peut-elle affecter le temps de réponse du vérin (ou du pilote) lors d'un changement de cap ?

DOCUMENT RESSOURCE – L'essentiel sur LE BUS et LE PROTOCOLE I2C

Présentation :

I2C (Inter Integrated Circuit) est un bus série permettant de transmettre des informations de façon asynchrone entre divers circuits connectés sur le bus. Le protocole de liaison est du type MAITRE/ESCLAVE. Chaque circuit est reconnu par son adresse et peut être soit transmetteur (émetteur) soit receveur (récepteur) de l'information.

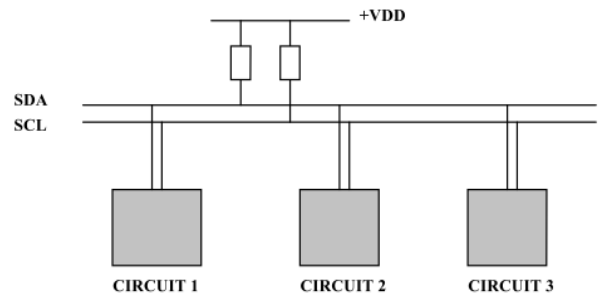
Deux fils SDA (Serial Data) et SCL (Serial Clock) véhiculent les informations entre les différents circuits. SDA et SCL sont des lignes bidirectionnelles, connectées au +VDD par l'intermédiaire de 2 résistances de tirage. Quand le bus est libre, c'est à dire quand il n'y a pas de transfert de données, les deux lignes sont à l'état haut (1 logique).

Pour que le transfert d'un bit soit valide, la donnée sur la ligne SDA doit être stable pendant que l'horloge est à l'état haut sur SCL. La valeur de SDA ne peut changer que si SCL est à l'état bas.

Un message débute par une condition de START et se termine par une condition de STOP.

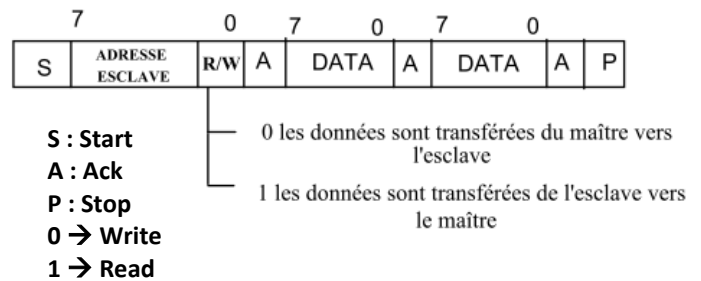
Condition de START : Passage de l'état haut à l'état bas de SDA pendant que SCL est à l'état haut.

Condition de STOP : Passage de l'état bas à l'état haut de SDA pendant que SCL est à l'état haut.



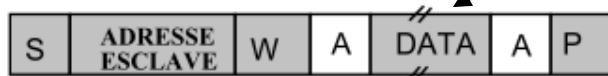
Transfert de données sur le bus :

Les données sont transférées par paquets de 8 bits (1 octet). Chaque octet doit être suivi d'un bit de reconnaissance « Acknowledge ». Dans le premier octet transféré, les 7 premiers bits représentent l'adresse de l'esclave et le 8^{ème} est le bit de Read/Write.

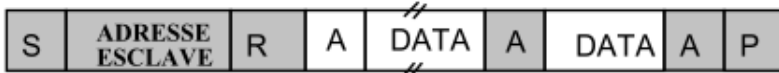


Exemple de trames transférées :

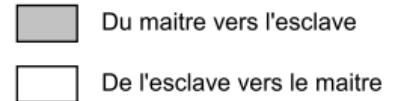
Mode maître transmetteur



Mode maître receveur



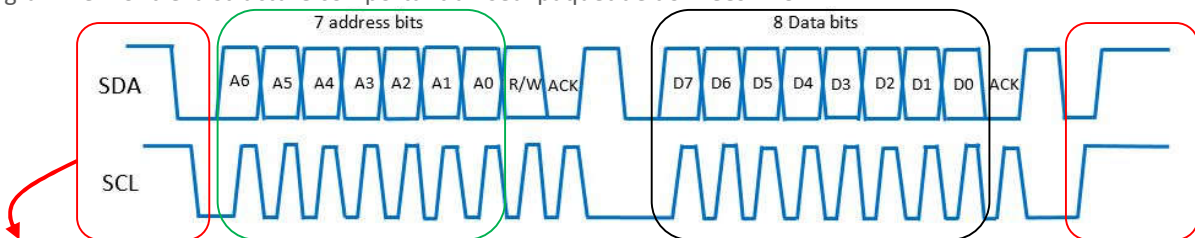
Plusieurs DATA sont possibles selon les circuits utilisés



Les signaux :

Les vitesses du bus I²C vont de 100 kbit/s en mode Standard, 400 kbit/s en mode Rapide, 1 Mbit/s en mode Rapide plus et 3,4 Mbit/s en mode Haute Vitesse. Chaque périphérique sur le bus est reconnu par une adresse unique à 7 ou 10 bits. Les données sont transférées dans des «paquets», qui incluent l'adresse de l'appareil, une commande de lecture / écriture, des accusés de réception et les données transférées.

Le diagramme montre la structure comportant un seul paquet de données I²C.



Au **début du paquet**, un appareil maître prend le contrôle du bus en baissant SDA tandis que SCL reste haut. Cela indique qu'un message va suivre. Ensuite, une **adresse de 7 (ou 10) bits** est transmise suivie d'un **bit R / W** pour indiquer s'il s'agit d'une instruction de lecture (1) ou d'écriture (0).

Le dispositif esclave adressé transmet ensuite un **bit d'accusé de réception (ACK)** en tirant la ligne SDA vers le bas. Si la ligne reste élevée, le maître peut déduire que l'esclave n'a pas reconnu l'adresse et des mesures correctives doivent être prises.

Une fois l'adresse acquittée par l'esclave, le maître continue de générer l'horloge et, en fonction du bit R / W, le maître ou l'esclave enverra des **données** sur le bus. Après chaque octet de données envoyé, un **ACK** est généré par le périphérique récepteur.

La **fin de paquet** est reconnue par la ligne SDA allant de bas à haut lorsque le SCL est déjà élevé.