

<b>Activités</b>						Support : Robot Mimi							
Manipulations		TD		Evaluation						Durée : 2h			
Compétences à acquérir													
A- Analyser			B- Modéliser				C- Expérimenter			D- Communiquer			
A1	A2	A3	B1	B2	B3	B4	C1	C2		D1	D2		
<i>Matériel à disposition :</i> <ul style="list-style-type: none"> <li>▪ Robot MIMI</li> <li>▪ Tablette avec applis MIMI</li> <li>▪ Oscilloscope numérique</li> </ul>							<i>Documents à disposition :</i> <ul style="list-style-type: none"> <li>▪ Dossier technique</li> <li>▪ Cours sur les bus CAN, RS232 et I<sub>2</sub>C</li> <li>▪ Document sur les servomoteurs</li> <li>▪ Aide à l'utilisation des applications</li> </ul>						

**Problématique n°1** : Quels sont les différents protocoles de communication au sein du robot MIMI ?

### Activité n°1 : Contrôle de la transmission des données sur la liaison « bluetooth »

Introduction (la liaison bluetooth) :

- Portée d'une dizaine de mètres en intérieur, modulation GFSK (Gaussian Frequency Shift Keying), vitesse 2.1 Mbit/s (pour le 2.0). Liaison facile à mettre en œuvre et de faible coût. **Attention en sortie du récepteur bluetooth, la liaison se comporte comme une RS232 classique à 57600 baud (1bit de start, 8 bits de données (LSB en tête), 1 bit de stop).**

- Le signal est considéré à « 0 » si  $V_{IL} < 0.8V$  et à « 1 » si  $V_{IH} < 2V$

Préparation :

- **Retrouver** dans le dossier technique, la trame de communication (tablette -> Interface bluetooth) correspondant à la transmission d'une commande GES\_PATTE (émission bus CAN). Le tableau ci-dessous est complété et correspond à :
  - pas d'inversion des pattes,
  - articulation coude sur patte centrale côté gauche uniquement sélectionné,
  - consigne de position = 486.
- **Vérifier et justifier** les valeurs de ce tableau.

#### Trame de communication : Tablette → Carte interface

Octet :	0	1	2	3	4	5	6	7	8	9	10	11	12
Valeur :	85	53	132	8	0	4	65	230	0	0	0	0	65
(hexa)	55	35	84	08	00	04	41	E6	00	00	00	00	41

Mesures :

- **Relever** les signaux PT\_RX (J4) et PT\_TX (J5) après la connexion de la tablette au robot.
- **Vérifier** la conformité des niveaux électriques (voir introduction) et **conclure**.
- **Vérifier** la durée d'un bit, **en déduire** la vitesse de transmission. Sachant que cette transmission ne tolère que 2% d'erreur, **conclure** ?

- **Relever** les signaux PT\_RX (J4) et PT\_TX (J5) lors de l'émission d'une commande de mouvement sur la patte centrale gauche et l'articulation « coude » (1 seule patte sélectionnée).
- **Vérifier** la conformité des niveaux électriques (voir introduction) et **conclure**.
- **Vérifier** la conformité des données transmises.
- **Vérifier** que les données transmises sont conformes à celles établies lors de la préparation (voir tableau).

## Activité n°2 : Contrôle de la transmission des données sur le bus CAN (réseau carte)

### Préparation :

- **Rappeler** rapidement les caractéristiques d'une liaison CAN (longueur, vitesse de transmission, intérêt ...).
- A l'aide du dossier technique, **relever** les identificateurs des trames acceptées par l'interface de communication bluetooth (émission vers la tablette).

Ci-dessous la trame CAN correspond aux conditions suivantes :

- alim = 12.3V,
- distance = 0,
- nb de clap = 1,
- version = 89,
- pas de servomoteur en défaut.

- A l'aide du dossier technique **vérifier** les valeurs dans ce tableau.

### Trame « Etat araignée »

	ID	DLC	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7	Data8
Valeur :	134	8	123	0	1	89	0	0	0	0
(hexa)	86	08	7B	00	01	59	00	00	00	00

### Mesures :

- **Retrouver** sur internet ou par tout autre moyen les niveaux de tension mini et maxi sur les lignes CAN\_H et CAN\_L du bus « CAN high speed ».
- **Relever** les signaux CAN\_L (J26) et CAN\_H (J25) après la mise sous tension du robot.
- **Vérifier** la conformité des niveaux électriques.
- **Vérifier** la durée d'un bit (état bas le plus court) puis **calculer** la vitesse de transmission. **Est-elle** conforme à celle donnée dans le dossier technique ?
- **Vérifier** la conformité des données transmises (voir tableau).

**Problématique n°2** : Les différentes communications sont-elles correctes (test et validation) ?

### Activité n°3 : Test et validation du réseau CAN

A partir du dossier technique (description des trames CAN : annexe 5), le tableau ci-dessous est compléter pour la commande « marche avant 3 pas »

Commande gestion mouvement :				
Identificateur CAN :		132		
Octet :	Donnée :	Valeur	Lsb	Remarque
0	COMMANDE	6	0	Marche
1	PARAMETRE 1	3	2 <sup>0</sup>	Nombre de pas
2	PARAMETRE 2	0	2 <sup>0</sup>	Sens (avant)

#### Expérimentation :

- A l'aide de l'application de gestion des liaisons numériques, **effectuer** l'émission sur le bus CAN de la commande de gestion de mouvement précédemment définie. **Vérifier** le fonctionnement.
- A l'aide de l'application de gestion des liaisons numériques, **effectuer** la lecture de la trame d'état du robot puis **compléter** le tableau ci-dessous.

Etat araignée :				
		Adresse CAN : 160		
Octet :	Donnée :	Valeur	Lsb	Remarque
0	V_ALIM			
1	DIST_OBSTACLE			
2	NB_CLAP			
3	VERSION			
4	ETAT_H			
5	ETAT_M			
6	ETAT_L			
7	SPARE			

### Activité n°4 : Test et validation du réseau I<sub>2</sub>C

A partir du dossier technique (description des trames I2C en annexe 5), le modèle du tableau permet d'allumer les Leds D3, D5, D7, et D9 pour le côté gauche du robot.

Gestion des Leds :				
Nom du réseau I2C utilisé (côté gauche) :		carte n°1		
Octet :	Donnée :	Valeur	Lsb	Remarque
0	ADRESSE I2C	0x70	0	Adresse composant I <sub>2</sub> C et Sens = WRITE
1	PARAMETRE	0xAA	2 <sup>0</sup>	Allumage led par niveau bas

### Expérimentation :

- A l'aide de l'application de gestion des liaisons numériques, **effectuer** l'émission sur le bus I<sub>2</sub>C de la commande d'allumage des Leds précédemment définie. **Vérifier** le fonctionnement. **Attention : avant l'émission de la commande attendre que le robot passe en veille (toutes les Leds éteintes).**
- A l'aide de l'application de gestion des liaisons numériques, **effectuer** la lecture de l'état des leds puis **compléter** le tableau ci-dessous.

Etat des Leds D3 à D10 côté gauche				
Nom du réseau I2C utilisé : carte n°1				
Octet :	Donnée :	Valeur	Lsb	Remarque
0	ADRESSE I2C		0	
1	PARAMETRE		2^0	
Etat des Leds D11 à D14 côté droit				
Nom du réseau I2C utilisé : carte n°2				
Octet :	Donnée :	Valeur	Lsb	Remarque
0	ADRESSE I2C		0	
1	PARAMETRE		2^0	

### Activité n°5 : Test et validation du réseau servo

A partir de la documentation technique des servos AX12 et du dossier technique le modèle du tableau ci-dessous permet l'écriture de la position de consigne 300 pour le servo qui pilote le coude de la patte centrale du côté gauche.

Gestion des servomoteurs (trame de commande) :				
Ecriture position de consigne :				
Octet :	Donnée :	Valeur	Lsb	Remarque
0	HEADER_1	0xFF	-	En-tête
1	HEADER_2	0xFF	-	En-tête
2	ID_SERVO	14	2^0	Identificateur
3	LG_TRAME	5	2^0	Taille trame
4	INSTRUCTION	3	2^0	Ecriture donnée
5	PARAM_1	0x1E	2^0	Adresse position de consigne
6	PARAM_2	0x2C	2^0	Poids faible position
7	PARAM_3	0x01	2^8	Poids fort position
8	CHECKSUM	0x9E	2^0	Complément somme de vérification sur 8 bits

Expérimentation :

- A l'aide de l'application de gestion des liaisons numériques, **effectuer** l'émission sur le réseau de servo la commande définie précédemment pour la couche applicative. **Vérifier** le fonctionnement.
- A l'aide de l'application de gestion des liaisons numériques, **effectuer** la lecture de la position actuelle de l'épaule pour la patte avant droite.

Gestion des servomoteurs (trame de commande) :				
Demande de lecture position actuelle (épaule patte avant droite) :				
Octet :	Donnée :	Valeur	Lsb	Remarque
0	HEADER_1		-	
1	HEADER_2		-	
2	ID_SERVO		2^0	
3	LG_TRAME		2^0	
4	INSTRUCTION		2^0	
5	PARAM_1		2^0	
6	PARAM_2		2^0	
7	CHECKSUM		2^0	
Gestion des servomoteurs (trame de status) :				
Résultat de la lecture de la position actuelle (épaule patte avant droite) :				
Octet :	Donnée :	Valeur	Lsb	Remarque
0	HEADER_1		-	
1	HEADER_2		-	
2	ID_SERVO		2^0	
3	LG_TRAME		2^0	
4	ERROR		-	
5	PARAM_1		2^0	
6	PARAM_2		2^8	
7	CHECKSUM		2^0	