




Fiche guide 2	TS SI		P.P.E. Robot solaire	
Analyse - Conception	4h			
 Lycée Polyvalent PIERRE EMILE MARTIN	Mise en œuvre de la carte « moteurs » MD23			

Nom(s) :	Classe :	Groupe :
----------	----------	----------

Objectifs

A la fin de cette étude :

- Vous serez capable de commander la fréquence de rotation des moteurs du robot à réaliser.

Matériels : Carte SSI + carte moteur MD23 + Motoréducteurs.

Logiciel : CodeVision AVR + programme de test « MD23prof.rom ».

Documentation

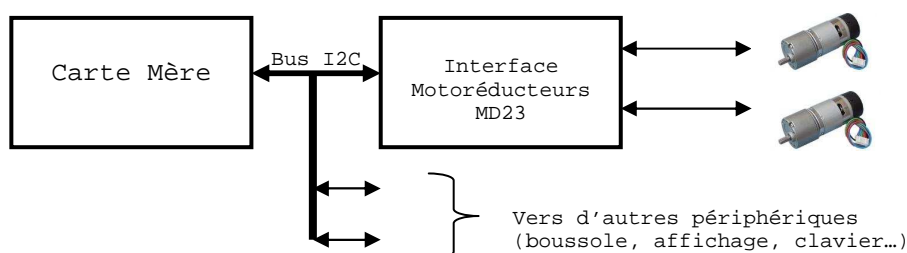
Fiche technique de la carte moteur MD23, schéma de la carte SSI.

Le présent document, la documentation technique et le programme de test sont téléchargeables sur le site WebGE à l'adresse <http://p.mariano.free.fr/> (rubrique PPE)

A) Mise en situation

Le « robot solaire » à réaliser se déplace grâce à deux **motoréducteurs** équipés de **codeurs**. Ces actionneurs sont commandés par une « **carte interface MD23** ». Cette carte est capable de gérer et de mesurer leur fréquence de rotation, de mesurer l'intensité du courant qui les traverse, de surveiller la tension d'alimentation, de contrôler leur accélération etc... Le nombre d'informations nécessaires au fonctionnement de la carte ajouté à celles qu'elle est capable de fournir justifie l'utilisation d'un **bus de communication série**. Le concepteur de cette carte a choisi le **bus I²C**.

Schéma fonctionnel partiel du robot solaire



Pour les essais, la carte mère est remplacée par la carte ATMELSSI. L'ensemble des structures matérielles étant réunies sur cette carte et sur la carte interface MD23, votre travail va se limiter à la **réalisation du logiciel** à implanter dans le microcontrôleur ATMEGA8535.

Pour cela, vous allez **créer et configurer un projet** avec le magicien du cross-compileur **CodeVisionAVR** afin d'obtenir la structure de votre programme.

	FG2	Mise en œuvre de la carte MD23		PPE ROBOT Solaire	1
---	------------	--------------------------------	---	-------------------	---

Vous complèterez ensuite cette structure avec les fonctions nécessaires à la mise en œuvre de l'interface MD23.

La suite de ce document décrit le travail à réaliser étape par étape. A la fin de cette activité, vous serez capable de commander les déplacements du robot.

B) Travail demandé

Etape 1 : Analyse de la fiche technique de la carte MD23

La commande de la carte MD23 s'effectue par l'intermédiaire d'un bus I²C. Si celui-ci simplifie son utilisation, il nécessite l'identification préalable des différentes fonctionnalités de la carte et les moyens d'accès à ces fonctionnalités (**adressage et mots de commande**). A ce point du projet, vous vous êtes familiarisé avec le bus I²C (abordé dans la FG1).

Les réponses aux questions qui suivent se trouvent dans la documentation de la carte MD23.

B1) Communication avec la carte MD23

Objectif : Paramétrer les sous-programmes assurant le transfert des informations entre la carte MD23 et la carte mère (lecture et écriture).

Lisez le paragraphe « **Controlling the MD23** »

Q1) A partir de quelles adresses peut-on accéder à la carte ? Quelle est l'adresse par défaut de la carte ? Comment peut-on la modifier ?

Les différentes fonctions de la carte sont commandées par l'intermédiaire de **registres** (mémoire 8 bits).

Q2) Combien de registre contient la carte MD23 ?

Les principales commandes ou informations accessibles sur la carte MD23 sont rassemblées dans le tableau du paragraphe « **Controlling the MD23** ». On accède à ces commandes ou à ces informations par l'intermédiaire de **registres**. L'accès à un registre s'effectue comme l'accès à un emplacement mémoire dans une EEPROM I²C de type ATMEL 24Cxx.

On donne ci-dessous un extrait de la documentation du logiciel CodeVision AVR.

Example how to access an Atmel 24C02 256 byte I2C EEPROM:

```
#define EEPROM_BUS_ADDRESS 0xa0

/* write a byte to the EEPROM */
void eeprom_write(unsigned char address, unsigned char data)
{
    i2c_start();
    i2c_write(EEPROM_BUS_ADDRESS);
    i2c_write(address);
    i2c_write(data);
    i2c_stop();
    delay_ms(10); /* 10ms delay to complete the write operation */
}
```



```

/* read a byte from the EEPROM */
unsigned char eeprom_read(unsigned char address)
{
    unsigned char data;
    i2c_start();
    i2c_write(EEPROM_BUS_ADDRESS);
    i2c_write(address);
    i2c_start();
    i2c_write(EEPROM_BUS_ADDRESS | 1);
    data=i2c_read(0);
    i2c_stop();
    return data;
}

```

Dans votre application, les sous-programmes d'accès aux registres de la carte MD23 (en lecture et en écriture) seront notés **MD23_WRITE** et **MD23_READ**. On donne ci-dessous les prototypes de ces fonctions.

```

// Prototype des fonctions s'accès à la carte MD23
// -----
unsigned char MD23_READ(unsigned char reg);
MD23_WRITE(unsigned char reg, unsigned char data);

```

Q3) La carte étant réglée sur l'adresse B0₍₁₆₎, on souhaite écrire la valeur 192₍₁₀₎ dans le registre speed1. **Complétez** ci-dessous l'appel à la fonction MD23_WRITE

MD23_WRITE(_____, _____);

Q4) La carte étant réglée sur l'adresse B0₍₁₆₎, on souhaite lire la valeur de la tension d'alimentation et la placer dans la variable « Tension ». **Ecrivez** ci-dessous la commande à placer dans le programme.

B2) Réglage de la vitesse de rotation des moteurs

Objectif : **Représenter** la caractéristique $vitesse_{(\%)} = f(Speed)$ dans différents modes de fonctionnement. **Identifier** les différents modes de fonctionnement de la carte MD23.

Lisez les paragraphes « **Mode Register** », « **Speed1 Register** » et « **Speed2/Turn Register** »

On suppose que la vitesse du robot est **proportionnelle** à la fréquence de rotation des roues. Nous pouvons donc raisonner en pourcentage (une commande à 0% correspond à une vitesse nulle, 100% à une vitesse maximum).

La fréquence de rotation des moteurs peut être réglée selon **4 modes**. Elle est proportionnelle à la valeur placée dans les registres 0 et 1 en mode 0 et 1.

Q5) A quel pourcentage de vitesse correspond la succession des ordres ci-dessous :

```

Mode ← 0
Speed1 ← 192
Speed2 ← 192

```

