



Documentation de la Carte Mère E = M6 2001.

Sommaire.

1. Présentation de la Carte Mère 2001	4
1.1. Introduction.	4
1.2. Présentation des différents périphériques.	5
1.2.1. Le Micro contrôleur.	5
1.2.2. La mémoire ROM.	6
1.2.3. La mémoire SRAM.	6
1.2.4. La mémoire FLASH.	6
1.2.5. La mémoire EEPROM.	6
1.2.6. Les Contrôleurs de moteurs.	6
1.2.7. Le « Peripheral port interface » (PPI).	6
1.2.8. L'horloge temps réel (HTR).	6
1.2.9. Le moniteur de tension.	8
1.2.10. La liaison série.	8
1.2.11. Le composant logique programmable (PALCE).	8
1.3. Le plan mémoire.	9
2. Définition des connecteurs d'entrée/sortie de la carte mère 2001.	10
2.1. Dialogue avec l 'extérieur du calculateur.	10
2.1.1. Connecteur d'alimentation.	10
2.1.2. Connecteurs DB9 de Liaison Série.	10
2.1.3. Connecteur DB9 « volant ».	11
2.1.4. Connecteurs USB de retour codeur des moteurs.	11
2.2. Dialogue interne au calculateur(CP, CL).	13
2.2.1. Connecteur entrées TOR.	13
2.2.2. Connecteur sorties TOR.	14
2.2.3. Connecteur PWM.	14
2.2.4. Connecteur commande moteur.	15
2.2.5. Connecteur de dialogue CM – CL.	15
3. Définition des Jumpers.	16
3.1.1. J10	16
3.1.2. J11	16
3.1.3. J12	16
3.1.4. J13	16
3.1.5. J20	16
3.1.6. J21	17
3.1.7. J22	17
3.1.8. J23	17
3.1.9. J71	17
3.1.10. J72	17
3.1.11. J81	17
3.1.12. J82	17
4. Suivis des différentes cartes mères.	18
4.1. Jade.	18
4.2. Léa.	18
5. La carte de test.	18

6. Vorace.	19
6.1. Introduction.	19
6.2. Origine.	19
6.3. Implantation sur la CM.	19
6.3.1. Développement et assemblage.	19
6.3.2. Programmation de l'EPROM.	19
6.4. Mise en œuvre.	19
6.5. Liste des commandes.	20
6.5.1. Help	20
6.5.2. Dir	20
6.5.3. Run	20
6.5.4. Download	20
6.5.5. Upload	21
6.5.6. Localisation	22
6.5.7. Jump	22
6.5.8. Dump de la mémoire externe.	22
6.5.9. Dump de la mémoire interne.	23
6.5.10. Intm ? ?	23
6.5.11. Edition de la mémoire externe	23
6.5.12. Clear Memory	24
6.5.13. Erase Flash	24
6.6. Travaux à effectuer afin d'obtenir une version définitive de VORACE.	24
7. Objectifs de la deuxième génération de calculateur pour le concours E=M6.	25
7.1. Introduction	25
7.2. Le calculateur de manière générale	25
7.3. La carte de puissance	26
7.4. Conclusion	27
AnnexeS :	28
Mise par écrit des modifications à apporter à la CP de Samir afin de tenir compte des différentes reprises filaires.	29
Schéma de la CM 2001 intégrant la correction filaire.	30
Top du projet.	30
Micro contrôleur.	31
Interface mémoire.	32
HTR & I/O.	33
Asservissement moteurs	34
Interface série (RS232)	35
Décodage.	36
Opto couplage des sorties TOR	37
Opto couplage des entrées TOR	38
Sorties PWM.	39
Connecteur d'extension.	40
Alimentation.	41
Source du composant logique programmable.	42

1. PRESENTATION DE LA CARTE MERE 2001

1.1. INTRODUCTION.

La carte mère 2001 a été conçue dans le but de donner au club une carte mère de dernière génération, plus puissante en terme de calcul et de capacité de commande. L'idée c'était d'avoir une carte de conception suffisamment souple pour permettre son utilisation plusieurs années de suite. Pour cela, elle devait être dotée de grosses capacités de commandes d'entrées / sorties. Les capacités retenues seront citées plus loin. Les grandes lignes du besoin du club étaient les suivantes :

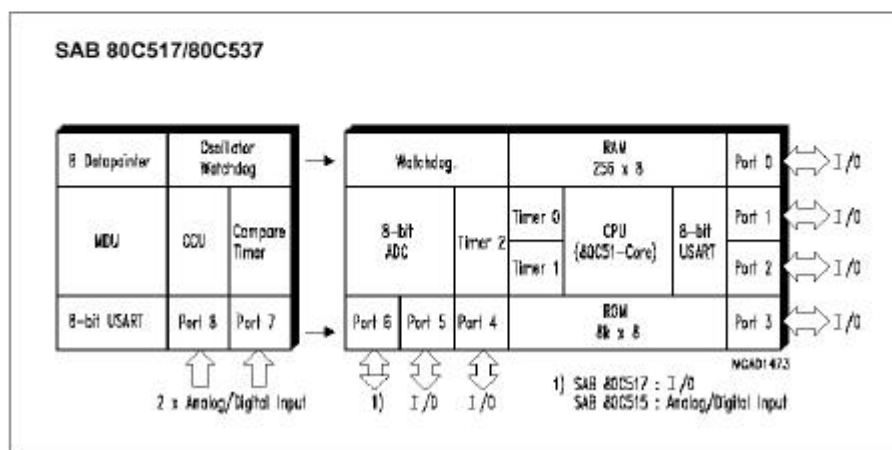
- Capacité de traitement d'une dizaine d'entrées TOR.
- Capacité à générer une dizaine de sorties TOR.
- Possibilité de commander trois moteurs DC avec asservissement numérique.
- Capacité avancée de gestion du temps.
- Grande capacité de stockage d'informations.
- Leds de debug pour la mise au point logiciel.
- Liaison série de dialogue avec une console.
- Capacité de téléchargement des programmes utilisateurs via une liaison série.
- Programme de boot par défaut capable d'initier un dialogue avec l'utilisateur et de mettre en œuvre les périphériques sans avoir à écrire de code.

1.2. PRESENTATION DES DIFFERENTS PERIPHERIQUES.

1.2.1. Le Micro contrôleur.

De la famille 8051, le contrôleur retenu est un 80C537 de chez Siemens (fondeur : Infineon). Il possède de nombreuses fonctionnalités qui sont énumérés dans l'extrait de documentation ci-dessous.

- Versions for 12 MHz and 16 MHz operating frequency
- 8 K x 8 ROM (SAB 80C517 only)
- 256 x 8 on-chip RAM
- Superset of SAB 80C51 architecture:
 - 1 μs instruction cycle time at 12 MHz
 - 750 ns instruction cycle time at 16 MHz
 - 256 directly addressable bits
 - Boolean processor
 - 64 Kbyte external data and program memory addressing
- Four 16-bit timer/counters
- Powerful 16-bit compare/capture unit (CCU) with up to 21 high-speed or PWM output channels and 5 capture inputs
- Versatile "fail-safe" provisions
- Fast 32-bit division, 16-bit 2 multiplication, 32-bit normalize and shift by peripheral MUL/DIV unit (MDU)
- Eight data pointers for external memory addressing
- Fourteen interrupt vectors, four priority levels selectable
- 8-bit A/D converter with 12 multiplexed inputs and programmable ref. voltages
- Two full duplex serial interfaces
- Fully upward compatible with SAB 80C515
- Extended power saving modes
- Nine ports: 56 I/O lines, 12 input lines
- Two temperature ranges available:
 - 0 to 70 °C
 - 40 to 85 °C
- Plastic packages: P-LCC-84, P-MQFP-100-2



1.2.1.1. Fréquence du microcontrôleur.

Le choix de la fréquence du μ c répond à des critères bien précis. En effet, dans le cas des contrôleurs de la famille 8051, la vitesse de la liaison série dépend complètement de la fréquence du contrôleur. Ici, il a fallu trouver une fréquence qui réponde à trois critères.

- Respect des données constructeur du 80C537 à savoir une fréquence maximum de 16 Mhz.
- Fréquence la plus élevée possible afin d'avoir une puissance de calcul maximal.
- Compatibilité entre la fréquence contrôleur et la fréquence de communication.

Le meilleur dénominateur commun aux trois contraintes ci-dessus est la fréquence de 14,xxx Mhz qui permet d'obtenir une fréquence de communication de 38400 bauds.

1.2.2.La mémoire ROM.

La carte mère possède une mémoire de type EPROM 32Ko dont le but est de stocker le programme de boot VORACE qui sera détaillé plus loin dans ce document.

1.2.3.La mémoire SRAM.

Présente sous la forme de deux pages de 16 Ko, Cette mémoire vive est destinée au programme exécuté par le micro contrôleur pour y stocker ces variables volatiles.

1.2.4.La mémoire FLASH.

32Ko de flash sont implémentés sur la CM pour le téléchargement des programmes utilisateurs via la liaison série. La mémoire de type Flash permet de garantir l'intégrité des données qui y sont stockées.

1.2.5.La mémoire EEPROM.

Présente sous la forme de deux pages de 16 Ko, Cette mémoire vive est destinée au programme exécuté par le contrôleur pour y stocker ces variables non volatiles.

1.2.6.Les Contrôleurs de moteurs.

Classique et bien connu, ce sont des LM629 qui sont implémentés. Au nombre de trois, leurs but est de commander par exemple deux moteurs de roues et un moteur de tir ou de ramassage de balle.

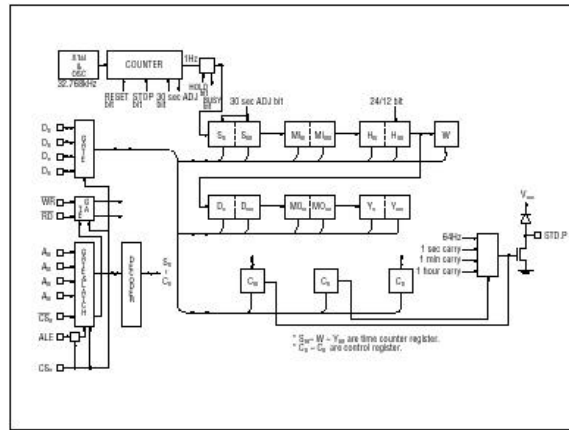
1.2.7.Le « Peripheral port interface » (PPI).

Ce composant a pour but de déporter le contrôle des entrées / sorties afin de libérer le processeur de cette tâche. Tel qu'il est configuré, le PPI contrôle 12 entrées et 12 sorties.

1.2.8.L'horloge temps réel (HTR).

Son but est de permettre au processeur de gérer le temps. En particulier, ça donne la possibilité de connaître le temps écoulé depuis le début du match. Toutefois, ce composant permet également de générer des I/O à intervalle de temps régulier à destination du processeur afin de lui éviter de devoir gérer des timers ou autres boucles d'attente.

Son organisation interne est celle-ci :



Et voici l'organisation des registres internes à l'horloge. Pour plus de détail, se référer au data sheet du composant qui explique sa mise en œuvre et son utilisation.

Address Input	A3	A2	A1	A0	Register Name	Data				Count value	Description
						D3	D2	D1	D0		
0	0	0	0	0	S1	S8	S4	S2	S1	0 to 9	1-second digit register
1	0	0	0	1	S10	*	S140	S20	S10	0 to 5	10-second digit register
2	0	0	1	0	MI1	mi8	mi4	mi2	mi1	0 to 9	1-minute digit register
3	0	0	1	1	MI10	*	mi40	mi20	mi10	0 to 5	10-minute digit register
4	0	1	0	0	H1	h8	h4	h2	h1	0 to 9	1-hour digit register
5	0	1	0	1	H10	*	PM/AM	h20	h10	0 to 2 or 0 to 1	PM/AM, 10-hour digit register
6	0	1	1	0	D1	d8	d4	d2	d1	0 to 9	1-day digit register
7	0	1	1	1	D10	*	*	d20	d10	0 to 3	10-day digit register
8	r	0	0	0	MO1	mo8	mo4	mo2	mo1	0 to 9	1-month digit register
9	r	0	0	1	MO10	*	*	*	MO10	0 to 1	10-month digit register
A	r	0	1	0	Y1	y8	y4	y2	y1	0 to 9	1-year digit register
B	r	0	1	1	Y10	y80	y40	y20	y10	0 to 9	10-year digit register
C	r	1	0	0	W	*	w4	w2	w1	0 to 6	Week register
D	r	1	0	1	C0	30 sec. ADJ.	IRQ FLAG	BUSY	HOLD	—	r
E	r	1	1	0	CE	t1	t0	ITRPT /STND	MASK	—	r
F	r	1	1	1	CF	TEST	24/12	STOP	REST	—	r

0 = "L" level, 1 = "H" level
 REST = RESET
 PM/AM = 1/0
 ITRPT/STND = INTERRUPT/STANDARD

- Notes:**
- 1) The writing of bit * is at discretion, but it is handled as '0' in the internal. In addition, it is unconditionally held at '0' during a read.
 - 2) The writings of '1' to IRQ FLAG bit, and '0' and '1' to BUSY bit are at discretion, but they are not carried out. The reading can be done. The writing of '0' to the IRQ FLAG bit is carried out.
 - 3) The bits except bit * and the BUSY bit can fully be read and written. However, the writing to the IRQ FLAG is effective for '0' only.
 - 4) PM/AM bit is 1 at PM and 0 at AM.

1.2.9. Le moniteur de tension.

Son but est de surveiller la tension de sortie du convertisseur DC/DC afin de resetter la CM dans le cas où le +5V s'écroulerait. Ce reset, permet d'éviter les actions délirantes des processeurs en cas de sous-alimentation. De plus, ce moniteur permet d'ajouter deux resets manuels, le premier implanté sur la CM et le second implanté en façade du robot.

1.2.10. La liaison série.

Au nombre de deux et géré par le micro contrôleur, ces liaisons ressortent de la CM via des DB9 au brochage normalisé. La première liaison série a pour vocation première de se connecter à un terminal afin d'établir le dialogue entre Vorace le logiciel implanté sur la CM et le terminal.

1.2.11. Le composant logique programmable (PALCE).

Ce composant implémenté sur la CM a deux utilités décrites ci-dessous.

1.2.11.1 Le décodage d'adresse.

Tout d'abord, il réalise le décodage d'adresse des différents périphériques qui composent la CM et qui sont cités plus haut. En annexe se trouve le source qui a permis de programmer ce composant. Il est donc possible en l'étudiant de retrouver les adresses des différents périphériques ainsi que les affectations des différentes broches.

1.2.11.2 La bufferisation.

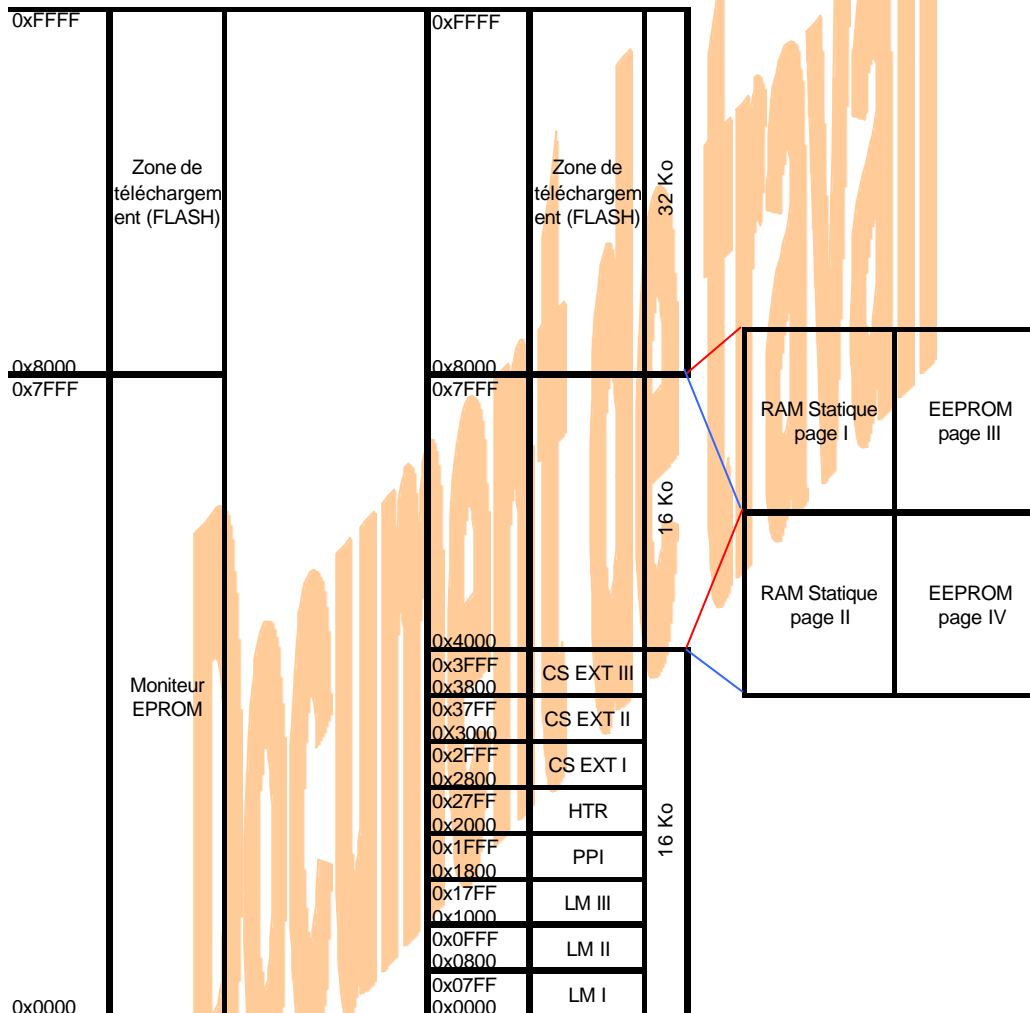
De plus, ce composant permet de bufferiser les signaux de contrôle du bus micro contrôleur qui ont un Fan out important (`_Rd` et `_WR`).

1.2.11.3 La programmation du composant.

La programmation de ce composant nécessite un programmeur qui supporte la programmation des composants PALCE22V10. Le fichier de programmation (Robot.jed) joint au dossier de la CM permet de réaliser cette programmation sans remarques particulières.

1.3. LE PLAN MEMOIRE.

Le tableau ci-dessous, permet de connaître l'organisation des périphériques à l'intérieur du plan mémoire du processeur. Le famille de contrôleur 8051 possède deux plans mémoire. Le premier est le plan mémoire réservé aux programmes. En principe, il n'est pas possible d'y toucher durant le fonctionnement du contrôleur, toutefois, j'ai contourné cette interdiction afin de permettre le téléchargement de programmes et leurs stockages en mémoire de programme par le contrôleur lui-même (Code auto morphique). Le second plan mémoire est appelé mémoire externe mais peut contenir aussi bien de la mémoire que des périphériques, ce qui est notre cas.



Les cinq première plages d'adresse de 0x0000 à 0x3FFF n'appellent pas de commentaires particuliers. Les CS I, II & III sont des segments de mémoires pré décodés pour de futures applications tel que la carte de localisation. Chacun des 8 premiers segments fait 2Ko(2048 octets).

Sur la plage contiguë à cette zone, se trouve la mémoire paginée. Sur un segments de 16 Ko, se trouve mappée 2 pages de 32 Ko de SRAM et 32 Ko d'EEPROM.

2. DEFINITION DES CONNECTEURS D'ENTREE/SORTIE DE LA CARTE MERE 2001.

2.1. DIALOGUE AVEC L'EXTERIEUR DU CALCULATEUR.

2.1.1. Connecteur d'alimentation.

➤ Nom du connecteur sur le circuit imprimé :

Ce connecteur est référencé J90 « Alim ».

➤ Position sur le circuit imprimé :

➤ Définition :

Ce connecteur à pour but d'alimenter la CM. Cette alimentation venant des batteries du robot, elle peut être comprise entre 18 et 36V. En fonctionnement nominale dans le robot et avec quatre batteries, cette tension est de 28.8V.

➤ Brochage :

1. Alimentation positive (+28.8V)

2. Masse batterie

2.1.2. Connecteurs DB9 de Liaison Série.

➤ Nom du connecteur sur le circuit imprimé :

La Carte Mère comporte deux liaisons séries. Et donc deux prise DB9 répondant à la norme Ad hoc. Les connecteurs sont les suivants : J50 « RS232_0 » et J51 « RS232_1 ».

➤ Position sur le circuit imprimé :

➤ Définition :

La CM est équipée de deux liaisons séries.

RS232_0 se connecte à un PC afin de dialoguer avec ce dernier au moyen de Vorace. Ainsi, depuis le PC et avec l'hyper Terminal, il est possible d'utiliser toutes les fonctionnalités de la CM.

RS232_1 à pour but de se connecter à un organe intelligent à l'intérieur du robot. Typiquement, avec la cellule optronique de Sick qui permet de mesurer les distances.

Fabricant :	Ref RS/Farnell :
-------------	------------------

➤ Brochage :

Le brochage de ces connecteurs est identique et standard. Il répond aux normes Ad hoc. C'est un connecteur de type DB9 Femelle.

- | | |
|----------|-------|
| 1. Masse | 6. NC |
| 2. RX | 7. NC |
| 3. TX | 8. NC |
| 4. NC | 9. NC |
| 5. NC | |

2.1.3. Connecteur DB9 « volant ».

➤ Nom du connecteur sur le circuit imprimé :

Sans identifiant.

➤ Position sur le circuit imprimé :

Entre la prise USB double et la DB9 de dialogue avec la console.

➤ Définition :

Cette DB9 Mâle ramène en façade du ordinateur les signaux qui sortent de la CM sur des borniers. On y retrouve le Reset externe, Le choix du mode de fonctionnement (Moniteur ou Développement), La visualisation de la présence « Tension batterie », La visualisation présence +5V en sortie du convertisseur DC/DC.

Fabricant :	Ref RS/Farnell :
-------------	------------------

➤ Brochage :

- | | |
|-------------------|------------------|
| 1. Relié à 6 & 4 | 6. Relié à 1 & 4 |
| 2. RX | 7. Relié à 8 |
| 3. TX | 8. Relié à 7 |
| 4. Relié à 1 & 6 | 9. NC |
| 5. GND Carte Mère | |

2.1.4. Connecteurs USB de retour codeur des moteurs.

2.1.4.1. Moteur 1.

➤ Nom du connecteur sur le circuit imprimé :

Ce connecteur est référencé J01 « Ret_moteur I & II ».

➤ Position sur le circuit imprimé :

Le retour codeur du moteur 1 s'effectue via la prise USB double type A. Cette prise étant à double étage, le retour codeur du LM629 1 s'effectue par la partie inférieure de la prise.

➤ Définition :

Le moteur 1 correspond au LM629 référencé U40. Sur le robot 2001 ce moteur se trouve du côté xxxxx du robot dans le sens de marche.

Sur ce connecteur USB se trouvent les signaux des codeurs intégrés au moteur(A et B) ainsi que les alimentations VCC et GND nécessaire au fonctionnement du codeur.

Fabricant :	Ref RS/Farnell :
-------------	------------------

➤ Brochage :

- | | |
|-------------------|-------------------|
| 1. VCC Carte Mère | 3. B |
| 2. A | 4. GND Carte Mère |

2.1.4.2.Moteur 2.

➤ Nom du connecteur sur le circuit imprimé :

Ce connecteur est référencé J01 « Ret_moteur I & II ».

➤ Position sur le circuit imprimé :

Le retour codeur du moteur 2 s'effectue via la prise USB double type A. Cette prise étant à double étage, le retour codeur du LM629 2 s'effectue par la partie supérieur de la prise.

➤ Définition :

Le moteur 2 correspond au LM629 référencé U41. Sur le robot 2001 ce moteur se trouve du coté yyyy du robot dans le sens de marche.

Sur ce connecteur USB se trouvent les signaux des codeurs intégrés au moteur(A et B) ainsi que les alimentations VCC et GND nécessaire au fonctionnement du codeur.

Fabricant :	Ref RS/Farnell :
-------------	------------------

➤ Brochage :

- | | |
|----------------|-------------------|
| 1. VCC Carte M | 3. B |
| 2. A | 4. GND Carte Mère |

2.1.4.3.Moteur3.

➤ Nom du connecteur sur le circuit imprimé :

Ce connecteur est référencé J02« Ret_moteur III ».

➤ Position sur le circuit imprimé :

Le retour codeur du moteur 3 s'effectue via la prise USB simple type A.

➤ Définition :

Le moteur 3 correspond au LM référencé U42. Sur le robot 2001 ce retour codeur n'est pas utilisé. Il a été prévu pour de futures extensions sur les robots des années suivantes.

Sur ce connecteur USB se trouvent les signaux des codeurs intégrés au moteur(A et B) ainsi que les alimentations VCC et GND nécessaire au fonctionnement du codeur.

Fabricant :	Ref RS/Farnell :
-------------	------------------

➤ Brochage :

3. VCC Carte Mère
4. A

5. B
6. GND Carte Mère

2.2. DIALOGUE INTERNE AU CALCULATEUR(CP, CL)

2.2.1. Connecteur entrées TOR.

➤ Nom du connecteur sur le circuit imprimé :

Ce connecteur est référencé J70 « Sortie TOR ».

➤ Position sur le circuit imprimé :

➤ Définition :

Ce connecteur reçoit les entrées TOR conditionnées par la CP.

Fabricant :	Ref RS/Farnell :
-------------	------------------

➤ Brochage :

1. Note « E1 »
2. Note « E1 »
3. Entrée 0
4. Entrée 1
5. Entrée 2
6. Entrée 3
7. Entrée 4
8. Entrée 5

9. Entrée 6
10. Entrée 7
11. Entrée 8
12. Entrée 9
13. Entrée 10
14. Entrée 11
15. Note « E2 »
16. Note « E2 »

Note « S1 » :

Note « S2 » :

2.2.2. Connecteur sorties TOR.

➤ Nom du connecteur sur le circuit imprimé :

Ce connecteur est référencé J80 « Entrée TOR ».

➤ Position sur le circuit imprimé :

➤ Définition :

Fabricant :	Ref RS/Farnell :
-------------	------------------

➤ Brochage :

1. Note « S1 »
2. Note « S1 »
3. Sortie 0
4. Sortie 1
5. Sortie 2
6. Sortie 3
7. Sortie 4
8. Sortie 5

9. Sortie 6
10. Sortie 7
11. Sortie 8
12. Sortie 9
13. Sortie 10
14. Sortie 11
15. Note « S2 »
16. Note « S2 »

Note « S1 » :

Note « S2 » :

2.2.3. Connecteur PWM.

➤ Nom du connecteur sur le circuit imprimé :

Ce connecteur est référencé J1B « PWM ».

➤ Position sur le circuit imprimé :

➤ Définition :

Fabricant :	Ref RS/Farnell :
-------------	------------------

➤ Brochage :

- | | |
|-------------------|--------------------|
| 1. VCC Carte Mère | 6. PWM 3 |
| 2. VCC Carte Mère | 7. PWM 4 |
| 3. PWM 0 | 8. PWM 5 |
| 4. PWM 1 | 9. GND Carte Mère |
| 5. PWM 2 | 10. GND Carte Mère |

2.2.4. Connecteur commande moteur.

- Nom du connecteur sur le circuit imprimé :

Ce connecteur est référencé J03 « Cmd_Moteur ».

- Position sur le circuit imprimé :

- Définition :

Ce connecteur fait transiter vers la CP les commandes d'activation ou de désactivation des sorties TOR.

Fabricant :	Ref RS/Farnell :
-------------	------------------

- Brochage :

- | | |
|-------------------|--------------------|
| 1. VCC Carte Mère | 6. Sign LM 2 |
| 2. VCC Carte Mère | 7. Magnétude LM 3 |
| 3. Magnétude LM 1 | 8. Sign LM 3 |
| 4. Sign LM 2 | 9. GND Carte Mère |
| 5. Magnétude LM 2 | 10. GND Carte Mère |

2.2.5. Connecteur de dialogue CM – CL.

- Nom du connecteur sur le circuit imprimé :

Ce connecteur est référencé J1A « DIN41612 64pts ».

- Position sur le circuit imprimé :

- Définition :

Sur ce connecteur transitent les signaux nécessaires au dialogue entre la CM et la CL. Par ce connecteur passent le bus du micro – contrôleur, les alimentations, des signaux PWM ainsi que les signaux analogique numérisable par le micro – contrôleur.

Fabricant :	Ref RS/Farnell :
-------------	------------------

➤ Brochage :

1A : VCC Carte Mère	17A : Réservé	1B : Bus d'adresse 15	17B : ADST
2A : VCC Carte Mère	18A : Interrupt Ext 1	2B : Bus d'adresse 14	18B : VA_GND
3A : VCC Carte Mère	19A : Interrupt Ext 2	3B : Bus d'adresse 13	19B : VA_REF
4A : Bus de données 7	20A : Interrupt Ext 3	4B : Bus d'adresse 12	20B : Canal analogique 0
5A : Bus de données 6	21A : Clock out	5B : Bus d'adresse 11	21B : Canal analogique 1
6A : Bus de données 5	22A : Reset	6B : Bus d'adresse 10	22B : Canal analogique 2
7A : Bus de données 4	23A : Reset inversé	7B : Bus d'adresse 9	23B : Canal analogique 3
8A : Bus de données 3	24A : T0	8B : Bus d'adresse 8	24B : Canal analogique 4
9A : Bus de données 2	25A : T1	9B : Bus d'adresse 7	25B : Canal analogique 5
10A : Bus de données 1	26A : T2	10B : Bus d'adresse 6	26B : Canal analogique 6
11A : Bus de données 0	27A : T2EX	11B : Bus d'adresse 5	27B : Canal analogique 7
12A : Chip Select Ext 1	28A : PWM 6	12B : Bus d'adresse 4	28B : Canal analogique 8
13A : Chip Select Ext 2	29A : PWM 7	13B : Bus d'adresse 3	29B : Canal analogique 9
14A : Chip Select Ext 3	30A : GND Carte Mère	14B : Bus d'adresse 2	30B : Canal analogique 10
15A : Signal de contrôle WR	31A : GND Carte Mère	15B : Bus d'adresse 1	31B : GND Carte Mère
16A : Signal de contrôle RD	32A : GND Carte Mère	16B : Bus d'adresse 0	32B : GND Carte Mère

3. DEFINITION DES JUMPERS.

Les jumpers situés sur le Carte Mère permettent de configurer sont fonctionnement. Les titres 1.3.x ci-dessous décrivent l'utilité et la mise en œuvre des différents jumpers qui composent la Carte Mère.

3.1.1.J10

3.1.2.J11

3.1.3.J12

3.1.4.J13

3.1.5.J20

[3.1.6.J21](#)

[3.1.7.J22](#)

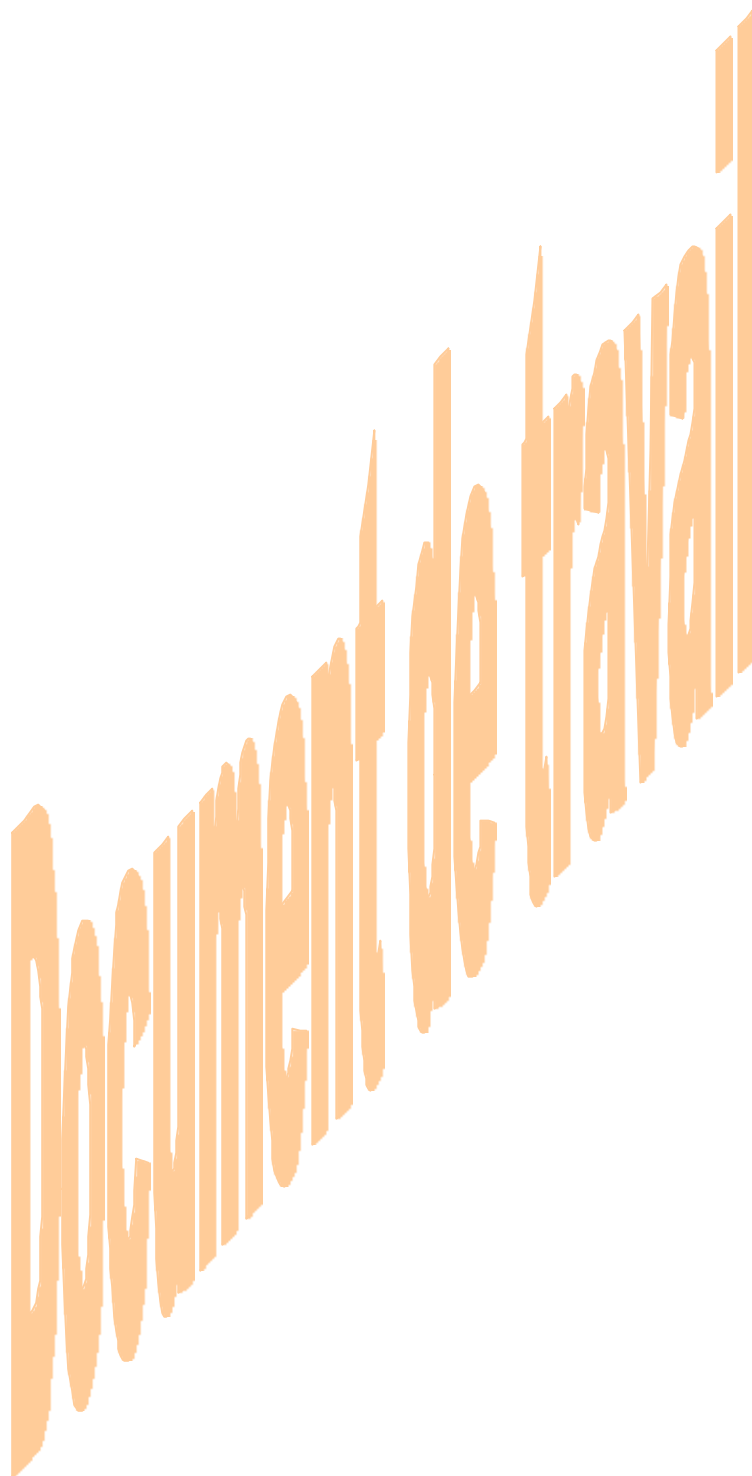
[3.1.8.J23](#)

[3.1.9.J71](#)

[3.1.10.J72](#)

[3.1.11.J81](#)

[3.1.12.J82](#)



4. SUIVIS DES DIFFERENTES CARTES MERES.

Pourquoi donner un nom aux différentes cartes. Ce fut un point à l'origine de nombreuses polémiques dans lesquels je ne rentre pas. Ce qu'il faut savoir, c'est que ces deux cartes équivalentes d'un point de vue électrique, ont vécu des choses différentes et ainsi peuvent être plus ou moins fragile. C'est pourquoi il est intéressant de pouvoir retracer leur vie afin de mieux apprécier leur fiabilité. Pour cela, il y avait deux solutions : soit je leur donnait un numéro soit je leur donnait un nom. Mon choix s'est porté sur les noms car je trouve cela plus sympas, plus facile à retenir et de plus, c'est pour moi un hommage aux personnes qui portent ou porteront ces prénoms.

4.1. JADE.

Première a avoir été réalisé, elle a essuyé les plâtres et subis la phase de mise au point. Jade a également souffert lors d'un essai d'intégration avec l'ancienne CP non documenté. Le résultat a été un court-circuit qui a détruit son processeur, son Horloge Temps Réel, sa mémoire SRAM et un démux 3 vers 8. Depuis, elle a été réparé et est opérationnel. Son rôle est de servir lors des essais afin de protéger Léa spécifiquement destiné au concours. Toutefois, en cas de défaillance de Léa au moment du concours ou d'une présentation (Murphy oblige !!) c'est Jade qui a pour rôle de la remplacer en attendant que Léa soit dépanné.

4.2. LEA.

Deuxième a avoir été réalisé, elle a été équipé par Pierre et testé par moi-même. Carte de référence, elle a été tenue à jours des évolutions de Jade. Léa est tout spécifiquement destiné à équipé le calculateur du concours. Protégé et bien traité, elle a la fiabilité maximal que l'on peut attendre que carte électronique. Dès son test finis, elle a subis une phase de « Burning » dont le but était d'éprouver sa fiabilité. Pour cela, elle a tourné nuit et jour pendant trois semaines avec un programme de teste dont le but été de faire travailler chaque composants qui compose la carte. Ceci ayant pour but de déceler les défaillance du aux défauts de jeunesse des composants. Ceci a permis de mettre en évidence une faiblesse de la première mémoire flash qui avait été intégré dessus.

Depuis, Léa fonctionne sans problème.

5. LA CARTE DE TEST.

Afin de pouvoir tester rapidement et avec le meilleur taux de couverture possible les différentes cartes mère, j'ai réalisé une carte de test qui une fois relié a la carte mère permet d'en tester le bon fonctionnement. Typiquement, la carte de test permet de vérifier le fonctionnement des 12 entrées TOR et des 12 sorties TOR. Pour cela, la carte de test est équipée de 12 jumpers qui permettent de fixer un état quelconque sur chacune des entrées TOR. Ainsi, il est possible avec Vorace de vérifier que l'on retrouve bien le niveau attendu. De cette manière, je valide toute la chaîne d'acquisition (Opto, PPI, buffer, contrôleur). De la même manière, je vérifie le bon fonctionnement des sorties TOR avec 12 leds montées sur la carte de test. Avec Vorace, j'active les sorties et je peux ainsi en vérifier le bon fonctionnement. De cette manière, je valide toute la chaîne de sortie (Opto, PPI, buffer, contrôleur).

Des systèmes équivalents permettent de mettre en œuvre les PWM destinés aux servo moteurs (afin de valider le contrôleur et le buffer associé) ainsi que le contrôle moteur avec les LM629 (afin de vérifier le fonctionnement des LM629 et des buffers associés).

6. VORACE.

6.1. INTRODUCTION.

Vorace est le logiciel enfouis qui se trouve sur la CM 2001. Son but est d'initialiser la CM. Celle-ci dispose de deux modes de fonctionnement. Un mode développement qui lui permet de dialoguer avec un PC via la liaison série durant les phases de mise au point. Un mode combat qui lui permet d'exécuter le programme qui aura été mis au point pour le match.

L'intérêt principal de Vorace se trouve dans son utilisation en mode développement. En effet, Vorace permet de mettre en œuvre l'ensemble de la CM sans avoir à écrire de code. Depuis une simple console hyper terminal, l'utilisateur peut activer ou désactiver les sorties. Il peut également connaître l'état des entrées et dialoguer avec tout les périphériques qui se trouvent sur la CM(HTR¹, PPI², LM629³, PWM⁴, LS⁵, mémoire) enfin, l'utilisateur peut lancer des commandes qui lui permettrons d'exécuter des actions complexes tel que le téléchargement de code machine, l'effacement de la Flash et bien d'autres choses encore.

Lors du boot, le contrôleur exécute Vorace ce qui lui permet d'initialiser les différents sous ensemble qui compose la CM. Après quoi il teste le sélecteur de mode afin de savoir si il doit continuer à s'exécuter ou bien si il doit exécuter le programme de combat qui se trouve dans sa mémoire Flash.

6.2. ORIGINE.

Vorace est une version modifié de PAULMON II.

C'est Paul Stoffregen (http://www.ece.orst.edu/~paul/8051-goodies/pm2_docs/intro.html) qui est à l'origine de Paulmon II. Son but étant de réaliser un moniteur convivial pour les cartes de développement à base de 8031. Paulmon II est un Freeware toutefois Paul Stoffregen demande simplement soit respecté l'origine de Paulmon II, c'est pourquoi, dans le package de la carte mère se trouvera toute la documentation écrite par Paul Stoffregen. Pour ceux qui s'y intéresseraient, je les invites à aller sur l'URL ci-dessus où ils trouveront une documentation à jour ainsi que les dernières évolutions du moniteur.

6.3. IMPLANTATION SUR LA CM.

6.3.1. Développement et assemblage.

6.3.2. Programmation de l'EPROM.

6.4. MISE EN ŒUVRE.

La mise en œuvre de Vorace est très simple. Lors du démarrage de la CM, il s'exécute tout seul. En fonction du mode choisis, soit il exécute le programme utilisateur, soit il continue de s'exécuter. Dans ce dernier cas, il va tenter d'établir une communication avec une console(Hyper terminal par exemple). Toutefois, Vorace ne peut pas savoir à quel vitesse la console va tenter de dialoguer avec lui. C'est pourquoi Vorace est équipé d'un «Auto baud ». Cette fonctionnalité lui permet de détecter automatiquement la vitesse de communication de la console. Pour cela, lorsque la CM à booté et

¹ Horloge temps réel.

² Peripheral port interface. Composant qui gère les entrées/sorties.

³ Composant qui gère le fonctionnement des moteurs.

⁴ MLI en Français(Modulation en largeur d'impulsion).

⁵ Liaison série.

qu'elle est connecté à la console, il suffit que cette dernière envoie le caractère Entrée(Return) pour que Vorace détecte la vitesse de communication. En attendant ce caractère, la CM restera muette. Une fois ce caractère reçu, Vorace montre son bon fonctionnement en affichant sur la console une page d'accueil. A ce moment là la communication bidirectionnelle entre la console et la CM est établie. Maintenant, la CM attend l'entrée d'une commande utilisateur.

6.5. LISTE DES COMMANDES.

Attention : Comme je vous l'ai dit plus haut, VORACE à pour origine Paulmon II c'est pourquoi dans la présentation des commandes ci-dessous vous verrez apparaître des références à Paulmon II lorsque je cite la documentation écrite par Paul Stoffregen.

6.5.1. Help

Keys assigned to this command : ?

6.5.1.1. Présentation

Cette commande permet de faire apparaître la liste des commandes à la disposition de l'utilisateur ainsi qu'un commentaire pour chacune d'elles. Ceci permet de se remémorer la touche qui permet de lancer chaque commande ainsi que les arguments attendus par cette dernière.

6.5.2. Dir

6.5.2.1. Présentation

6.5.2.2. Utilisation

6.5.3. Run

Keys assigned to this command : R

6.5.3.1. Extrait de la documentation de PaulmonII

Allows you to run programs which have a 64 byte header indicating they are programs. This provides a simple means to run a program without having to specify which memory location to jump to. For programs without the 64 byte header, use the Jump command. While the program is running, it has complete control over the processor... PAULMON2 is not still running. If the program crashes (infinite loop), the only way to return to PAULMON2 is to reset the processor.

6.5.4. Download

Keys assigned to this command : D

6.5.4.1. Extrait de la documentation de PaulmonII

The download command allows PAULMON2 to receive an Intel-Hex file, which is written into memory as it is received. After executing this command, you need to transmit your intel-hex file to

PAULMON2. Since the Intel-Hex is a text only format, the ascii-send feature in your terminal emulation program should work. A protocol such as xmodem or zmodem should not be used. The Intel-Hex format includes information specifying where the data is to be written into memory, so the memory pointer value is not used.

When downloading to RAM, PAULMON2 can accept data at the maximum possible rate, so it is not necessary to use character pacing or other delays. To abort a download, just press . As PAULMON2 receives each line of Intel-Hex data, it prints a period. When a download is finished or aborted, a brief summary is shown.

VORACE Loc:2000 > Download

Begin ascii transfer of Intel hex file, or ESC to abort

.....
.....
.....
.....

Download completed

Summary:
249 lines received
3923 bytes received
3923 bytes written
No errors detected

If the download command reports errors attempting to write, such as:

Summary:
246 lines received
3834 bytes received
630 bytes written
Errors:
3204 bytes unable to write

then it has correctly received the Intel-Hex, but the memory which was specified by the Intel-Hex file is read-only. In this example, many of the read-only memory locations contained the same data (because it was a new revision of the same code), so PAULMON2 verified 630 of the bytes as correctly written.

When PAULMON2 received a corrupted input, a summary similar to this will be printed:

Summary:
3 lines received
62 bytes received
62 bytes written
Errors:
1 bad checksums
2 unexpected begin of line
17 unexpected hex digits
2 unexpected non hex digits

[6.5.5.Upload](#)

Keys assigned to this command : U

6.5.5.1 [Extrait de la documentation de PaulmonII](#)

The upload command allows you to specify a portion of memory which is to be transmitted to your computer in Intel-Hex format. After the memory range is specified, PAULMON2 prompts for a keypress before transmitting, so you can activate a "capture" feature in your terminal emulation program to save the incoming data to disk. A typical upload may look like this:

VORACE Loc:0100 > Upload

First Location: 0000

Last Location: 003A

Sending Intel hex file from 0000 to 003A

Press any key:

:10000000208A60220037420116A2202200B742D1C

:10001000116A22022013020A72FFFF02201BFFFF57

:10002000FFFFFF0220230209AFFFF02202B216EFF

:0B003000016A0162215D217F218C012B

:00000001FF

6.5.6 [Localisation](#)

Keys assigned to this command : N

6.5.6.1 [Extrait de la documentation de PaulmonII](#)

This allow you to change the memory pointer:

VORACE Loc:8000 > New location

New memory location: 2C00

VORACE Loc:2C00 >

6.5.7 [Jump](#)

Keys assigned to this command : J

6.5.7.1 [Extrait de la documentation de PaulmonII](#)

The Jump command allows you to jump directly to a program. PAULMON2 pushes 0000 onto the stack, so a program which terminates with "RET" will rerun PAULMON2. If the program crashes (infinite loop), the only way to return to PAULMON2 is to reset the processor.

6.5.8 [Dump de la mémoire externe.](#)

6.5.8.1 [Extrait de la documentation de PaulmonII](#)

Keys assigned to this command : H

The Hex Dump command allows you to inspect the external memory. 256 bytes are printed in hexadecimal and ascii, beginning at the location of the memory pointer. The data is read from code memory (with MOVC).

VORACE Loc:2100 > Hex dump memory

```

2100: FD 31 3F 40 C6 CD 12 20 6A ED C0 E0 31 1D D0 E0 }1?@FM jm@`1 P`
2110: 2A FA E4 3B FB DC B4 C3 8A 82 8B 83 22 EB C4 54 *zd;{\4C "kDT
2120: F0 FB EA C4 54 0F 4B FB EA C4 54 F0 FA 22 EA C4 p{jDT K{jDTpz"jD
2130: 54 0F FA EB C4 54 F0 4A FA EB C4 54 0F FB 22 C3 T zkDTpJzkDT {"C
2140: C0 F0 94 30 F5 F0 94 0A 40 06 E5 F0 94 07 F5 F0 @p 0up @ ep up
2150: E5 F0 C3 54 F0 60 01 D3 E5 F0 D0 F0 22 C0 E0 C4 epCTp` SepPp"@`D
2160: 54 0F 24 F6 50 02 24 07 24 3A 11 6A D0 E0 C0 E0 T $vP $ $: jP`@`
2170: 54 0F 24 F6 50 02 24 07 24 3A 11 6A D0 E0 22 C0 T $vP $ $: jP`"@
2180: E0 E5 83 31 5D E5 82 31 5D D0 E0 22 C0 E0 E4 93 `e l]e l]P`"@`d
2190: A3 60 0A A2 E7 54 7F 11 6A 40 02 80 F1 D0 E0 22 #` "gT j@ qP`"
21A0: 90 2D 50 11 45 EF 31 5D EE 31 5D 11 06 75 83 10 -P Eo1]n1] u
21B0: F1 BB 50 2F 75 82 08 E4 93 B4 A5 23 05 82 E4 93 q;P/u d 4%# d
21C0: B4 64 1C C0 83 74 DA C0 E0 74 21 C0 E0 05 82 E4 4d @ tZ@`t!@` d
21D0: 93 C0 E0 05 82 E4 93 C0 E0 22 D0 83 12 20 06 05 @` d @`"P
21E0: 83 21 B0 90 2D 60 31 8C 11 62 11 16 75 F0 A0 C0 !0 -`1 b up @
21F0: F0 75 F0 21 C0 F0 F5 F0 90 10 00 F1 BB 50 21 75 pup!@pup q;P!u
    
```

[6.5.9.Dump de la memoire interne.](#)

[6.5.9.1.Extrait de la documentation de PaulmonII](#)

Keys assigned to this command : I

This command displays a dump of the internal memory, for example:

VORACE Loc:FFF0 > Hex dump internal memory

```

00: 00 00 00 00 43 03 F0 FF 55 55 55 55 AA AA AA AA
10: 55 55 55 55 AA AA AA AA 55 55 55 55 AA AA AA AA
20: 55 55 55 55 AA AA AA AA 55 55 55 55 AA AA AA AA
30: 55 A0 01 E6 07 0A 00 01 43 31 0F 48 0B 00 0B AA
40: 55 55 55 55 AA AA AA AA 55 55 55 55 AA AA AA AA
50: 55 55 55 55 AA AA AA AA 55 55 55 55 AA AA AA AA
60: 55 55 55 55 AA AA AA AA 55 55 55 55 AA AA AA AA
70: 55 55 55 55 AA AA AA AA E2 33 AA FF AA AA AA AA
    
```

This function can be useful to examine the internal memory after a user program has run. However, VORACE uses internal RAM, so the following memory ranges are likely to have been overwritten by VORACE, shown in red above.

Memory Range	Usage by PAULMON2
00 to 07	Registers R0 to R7
10 to 1F	Used during Download (Error counts)
31 to 40	Stack
78 to 7B	Baud rate storage for warm reset

[6.5.10.Intm ??](#)

[6.5.10.1Présentation](#)

[6.5.10.2Utilisation](#)

[6.5.11.Edition de la mémoire externe](#)

Keys assigned to this command : E

6.5.11.1 Présentation

Cette commande permet d'écrire directement dans la mémoire externe. Bien sûr, ceci inclut la mémoire mais aussi tous les périphériques tels que le PPI, les LM629, l'horloge temps réel, etc ...

6.5.12 Clear Memory

Keys assigned to this command : C

6.5.12.1 Présentation

La commande « Clear » est une solution simple pour remplir une zone mémoire de zéros. A la mise sous tension, les mémoires de type RAM sont bien souvent remplies de valeurs délirantes. Cette commande permet de se retrouver avec un contenu « Clean » afin de visualiser plus simplement les changements de contenus.

6.5.13 Erase Flash

6.5.13.1 Présentation

6.5.13.2 Utilisation

6.6 TRAVAUX A EFFECTUER AFIN D'OBTENIR UNE VERSION DEFINITIVE DE VORACE.

7. OBJECTIFS DE LA DEUXIEME GENERATION DE CALCULATEUR POUR LE CONCOURS E=M6.

7.1. INTRODUCTION

Le présent document représente le cahier des charges pour le développement du calculateur 2002. Il regroupe l'ensemble des actions qu'il serait nécessaire de mener à bien pour obtenir un calculateur répondant à notre besoin pour le prochain concours. Ces actions ont été extraites du bilan 2001 (Réunion + compte rendu) et de l'expérience du concours qui nous a montré les faiblesses et les points faibles de notre architecture. Ce Cdc⁶ se veut le plus exhaustif possible toutefois si vous aviez des idées à rajouter n'hésitez pas à m'en faire part. Je ne sais pas qui réalisera ces travaux mais ce document constituera toujours une base de travail pour le futur responsable du calculateur.

7.2. LE CALCULATEUR DE MANIERE GENERALE

- Le chargeur de batterie acheté lors de la coupe 2001 ruine l'un des intérêts de la carte de dispatch. Pour cette raison, les fonctions résiduelles de cette dernière pourraient être intégrées sur la carte de puissance.
- Au vu du coût des convertisseurs DC/DC, il paraît judicieux d'éviter d'en mettre sur la CM⁷ et la CL⁸. Par conséquent la prochaine version de la CM sera dépourvu de son propre convertisseur afin de centraliser ce dernier sur la CP⁹. C'est la CP qui, via le bus interne, alimentera la CM et la CL.
- Le gain apporté par la fusion des cartes CP & Dispatch, permettra de ce ramener à une configuration à deux cartes :
 - Carte Mère (sans alimentation propre mais avec la possibilité de la connectée sur une alimentation externe pour la mise au point) .
 - Carte Puissance résultat de la fusion CP et Dispatch avec en plus la gestion des alimentation 5V pour la CM et une éventuel futur carte de localisation.
- Ces modifications donnent ainsi la possibilité d'intégrer dans le futur la carte de localisation qui viendra ainsi compléter le calculateur.
- La connectique utilisé aujourd'hui pour relier les différentes cartes du calculateur étant trop gourmande en place, le futur calculateur seras équipé d'un bus interne qui passera sur chacune des cartes (CM, CP, CL) afin de faire transiter toutes les informations nécessaires au fonctionnement du système (Alimentations, dialogue CM-CP, dialogue CM-CL).
- La nouvelles CM verra ses I/O¹⁰ repensées avec la possibilité de passer à 24 I/O au lieu de 12.
- Les cartes du nouveau calculateur auront cinq colonnettes de fixation au lieu de quatre pour des questions de rigidité.
- Les cartes du nouveau calculateur intégreront des points de raccordement à la masse pour la mise au point.
- La nouvelle version du calculateur se connectera sur un fond de panier qui supportera la connectique nécessaire pour relier le calculateur à son environnement. C'est de ce fond de panier que partiront les fils qui iront sur les Batteries, Voyants, Capteurs et Actionneurs. (CF : schéma ci-dessous)

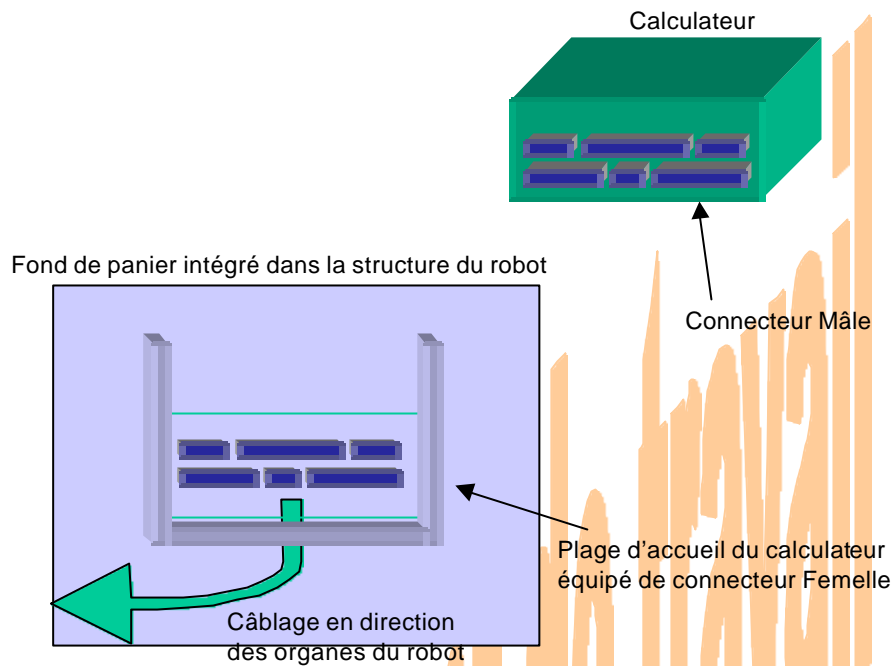
⁶ Cahier Des Charges

⁷ Carte Mère

⁸ Carte de Localisation

⁹ Carte de Puissance

¹⁰ Entrées / Sortie en anglais dans le texte ;-)



- Une réflexion devra être menée afin de spécifier rapidement les interfaces du calculateur. Ceci permettra d'anticiper la réalisation du fond de panier et du câblage. De même, les méca pourront tirer parti de ces informations pour la réalisation de la structure du robot (Structure globale, passage de câble, etc...) ainsi que du fond de panier qui supportera les connecteurs et qui devra probablement être réalisé en découpe laser.
- La réalisation d'une « maquette d'interface »¹¹ devra être envisagée afin d'accélérer la réalisation du fond de panier et la fixation du calculateur au sein du robot. De plus, cette maquette donnera un aperçu fidèle de l'encombrement afin que les méca puissent en tenir compte lors de leur design.
- La réalisation de cartes de tests devra être sérieusement envisagée afin que chaque carte du calculateur puisse être testée rapidement et avec un bon taux de couverture.
- Une prochaine réunion aura pour but de fixer des dates pour la réalisation et le suivi du calculateur.
- Dans la mesure du possible, il serait judicieux de prévoir la réutilisation des composants en stock ou montés sur cartes afin de réduire les coûts de développement.

7.3. LA CARTE DE PUISSANCE

- La nouvelle version de la CP intégrera le remplacement des deux régulateurs de tension 5V par les convertisseurs DC/DC choisis lors de la mise au point.
- La nouvelle version de la CP intégrera la mise en forme des signaux issus du capteur Sick si tant est que l'on en ait encore besoin.

¹¹ Boîtier du calculateur avec tous les circuits imprimés dans leur position finale mais seulement équipé de leurs connecteurs

- Une correction sera apporté afin de rajouter des résistance de Pull-Up¹² en sortie des composants d'isolations¹³ de la CP.
- Une correction sera apporté afin de remplacer les résistances en sortie du pont en H (composant qui commande les moteurs de déplacements) par les résistances de puissances choisies durant la mise au point.
- Afin de gagner de la place et de réduire le nombre de composant de la CP, les composants en logique discrète type 7400 ou 7406 qui réalise l'interface entre la CM et la commande des moteurs seront remplacés par un composant logique programmable¹⁴ (Un seul composant au lieu de trois).
- La possibilité d'implanter une diode de puissance sur le chemin d'alimentation des moteurs sera envisagée afin d'éviter que les perturbations générées par les moteurs ne remontent sur l'ensemble des alimentations du robot.
- Les alimentations et le blindage de la CP sera particulièrement soigné afin de se prémunir des problèmes de CEM.
- Dans la mesure du possible et du raisonnable, la CP fera appel aux composants CMS afin de réduire l'encombrement des fonctionnalités existantes.

7.4. CONCLUSION

Ce document constitue le premier volet d'une série de documents qui auront pour but d'assurer le transfert de connaissance qui a toujours manqué au club. Après avoir tant parlé de ce problème à nos sponsors, maintenant il s'agit de mettre en pratique nos belles paroles.

Dans ce document, aucun chapitre ne parle des corrections à apporter à la carte mère et pour cause, vous n'avez relevé aucun problème la concernant! Je crois même me rappeler vous avoir entendu dire quelle était parfaite ... ;-) Je plaisante bien sûr !

Cependant, je suis toujours à l'écoute de vos remarques afin d'étayer plus avant ce document.

¹² Résistance de tirage au +5V

¹³ Optocoupleurs

¹⁴ type PAL22V10



ANNEXES :

MISE PAR ECRIT DES MODIFICATIONS A APPORTER A LA CP DE SAMIR AFIN DE TENIR COMPTE DES DIFFERENTES REPRISES FILAIRES.

- Rajouter des résistances de tirages en sorties des opto-coupleurs.
- Remplacer la résistances $1\ \Omega\ \frac{1}{4}W$ en sortie des ponts en H de commandes des moteurs par une résistance de 30 W Référencé : xxx
- Modifier l 'étage de mise en forme du signal émanant du capteur Sick.
- Remplacer les régulateurs qui sont sur la CP par des convertisseurs DC/DC. En remplacement du régulateur d'alimentation de la CP, mettre un convertisseur de référence : xxx et en remplacement du régulateur d'alimentation des servo moteurs, mettre un convertisseur de référence : xxx.
- Rajouter du découplage.
- Revoir les alimentations du CI et en particulier le retour à la masse du pont en H.

SCHEMA DE LA CM 2001 INTEGRANT LA CORRECTION FILAIRE.

[Top du projet.](#)

Document de travail

Ce document est la propriété du club **ROBOTIK** Ingénieurs 2000 ; il ne peut être communiqué à des tiers et/ou reproduit sans l'autorisation préalable écrite du club **ROBOTIK** et son contenu ne peut être divulgué. © - Club ROBOTIK 2001

Micro contrôleur.

Document de travail

[Interface mémoire.](#)

Document de travail

HTR & I/O.

Document de travail

Ce document est la propriété du club **ROBOTIK** Ingénieurs 2000 ; il ne peut être communiqué à des tiers et/ou reproduit sans l'autorisation préalable écrite du club **ROBOTIK** et son contenu ne peut être divulgué. © - Club ROBOTIK 2001

Asservissement moteurs

Document de travail

Interface série (RS232)

Document de travail

Décodage.

Document de travail

Ce document est la propriété du club **ROBOTIK** Ingénieurs 2000 ; il ne peut être communiqué à des tiers et/ou reproduit sans l'autorisation préalable écrite du club **ROBOTIK** et son contenu ne peut être divulgué. © - Club ROBOTIK 2001

Opto couplage des sorties TOR

Document de travail

Opto couplage des entrées TOR

Document de travail

Sorties PWM.

Document de travail

[Connecteur d'extension.](#)

Document de travail

Alimentation.

Document de travail

SOURCE DU COMPOSANT LOGIQUE PROGRAMMABLE.

```
;PALASM Design Description
```

```
;----- Declaration Segment -----
```

```
TITLE ROBOT.PDS
PATTERN A
REVISION 1.0
AUTHOR D.AUCLAIR
COMPANY I2000
DATE 01/02/01
```

```
CHIP DECODER PAL22V10
```

```
;----- PIN Declarations -----
```

```
PIN 1      A15      COMBINATORIAL ; INPUT
PIN 2      A14      COMBINATORIAL ; INPUT
PIN 8      RESET    COMBINATORIAL ; INPUT
PIN 9      _Sel_Mem COMBINATORIAL ; INPUT
PIN 10     _WR      COMBINATORIAL ; INPUT
PIN 11     _RD      COMBINATORIAL ; INPUT
PIN 13     _PSEN    COMBINATORIAL ; INPUT
PIN 12     GND      ; INPUT

PIN 14     _B_WR    COMBINATORIAL ; OUTPUT
PIN 15     _B_RD    COMBINATORIAL ; OUTPUT
PIN 16     _GATE    COMBINATORIAL ; OUTPUT
PIN 18     _CE_ROM  COMBINATORIAL ; OUTPUT
PIN 19     _CE_EEPROM COMBINATORIAL ; OUTPUT
PIN 20     _CE_RAM  COMBINATORIAL ; OUTPUT
PIN 21     _CE_FLASH COMBINATORIAL ; OUTPUT
PIN 22     _OE_FLASH COMBINATORIAL ; OUTPUT
PIN 23     _CS_138  COMBINATORIAL ; OUTPUT
PIN 24     VCC      ; INPUT
```

```
;----- Boolean Equation Segment -----
```

```
EQUATIONS
```

```
;-----bufferisation des signaux Rd et Wr
```

```
_B_WR = _WR
_B_RD = _RD
```

```
;-----Decodage de la memoire programme (_PSEN=0)
```

```
;----Decodage de 0x0000 a 0x7FFF
```

```
_CE_ROM = /(A15 * /RESET * !_PSEN)
```

```
;----Decodage de 0x8000 a 0xFFFF
```

```
_CE_FLASH = /(A15 * /RESET)
```

```
;-----Ruse de sioux!
```

```
_OE_FLASH = /(A15 * (!_PSEN + !_RD) * /RESET)
```

```
;----- Gate du buffer de bus
```

```
_GATE = A15
```

```
;-----Decodage de la memoire de donnees (_PSEN=1)
```

```
; Debut de plage : 0x4000 - fin : 0x7FFF avec _PSEN = 1
```

```
:_CE_EEPROM = /(A15 * A14 * /RESET * !_Sel_Mem * _PSEN * (!_rd + !_wr))
```

```

_CE_EEPROM = /(A15 * A14 * /RESET * /_Sel_Mem * _PSEN)

; Debut de plage : 0x4000 - fin : 0x7FFF avec _PSEN = 1
:_CE_RAM = /(A15 * A14 * /RESET * _Sel_Mem * _PSEN * (/rd + /wr))
_CE_RAM = /(A15 * A14 * /RESET * _Sel_Mem * _PSEN)

; Debut de plage : 0x0000 - fin : 0x3FFF avec _PSEN = 1
:_CS_138 = /(A15 * /A14 * /RESET * _PSEN * (/rd + /wr))
_CS_138 = /(A15 * /A14 * /RESET * _PSEN)

;----- Simulation Segment -----
SIMULATION
TRACE_ON A15 A15 RESET _Sel_Mem _WR _RD _PSEN _B_WR _B_RD _CE_ROM _CE_EEPROM _CE_RAM
_CE_FLASH _OE_FLASH _CS_138

SETF A14 A15 RESET _Sel_Mem _WR _RD _PSEN
CHECK _B_WR _B_RD _CE_ROM _CE_EEPROM _CE_RAM _CE_FLASH _OE_FLASH _CS_138

SETF A14 A15 RESET _Sel_Mem /_WR _RD _PSEN
CHECK /_B_WR _B_RD _CE_ROM _CE_EEPROM _CE_RAM _CE_FLASH _OE_FLASH _CS_138

SETF A14 A15 RESET _Sel_Mem _WR /_RD _PSEN
CHECK _B_WR /_B_RD _CE_ROM _CE_EEPROM _CE_RAM _CE_FLASH /_OE_FLASH _CS_138

SETF A14 /A15 /RESET _Sel_Mem _WR _RD /_PSEN
CHECK _B_WR _B_RD /_CE_ROM _CE_EEPROM _CE_RAM _CE_FLASH _OE_FLASH _CS_138

SETF A14 /A15 /RESET _Sel_Mem _WR _RD _PSEN
CHECK _B_WR _B_RD _CE_ROM /_CE_EEPROM _CE_RAM _CE_FLASH _OE_FLASH _CS_138

SETF A14 /A15 /RESET /_Sel_Mem _WR _RD _PSEN
CHECK _B_WR _B_RD _CE_ROM _CE_EEPROM /_CE_RAM _CE_FLASH _OE_FLASH _CS_138

SETF A14 A15 /RESET _Sel_Mem _WR _RD /_PSEN
CHECK _B_WR _B_RD _CE_ROM _CE_EEPROM _CE_RAM /_CE_FLASH _OE_FLASH _CS_138

SETF A14 A15 RESET _Sel_Mem _WR _RD /_PSEN
CHECK _B_WR _B_RD _CE_ROM _CE_EEPROM _CE_RAM /_CE_FLASH _OE_FLASH _CS_138

TRACE_OFF

;-----

```