

Auteur :

Blaise Fracheboud

Questions :

info@hemisson.com

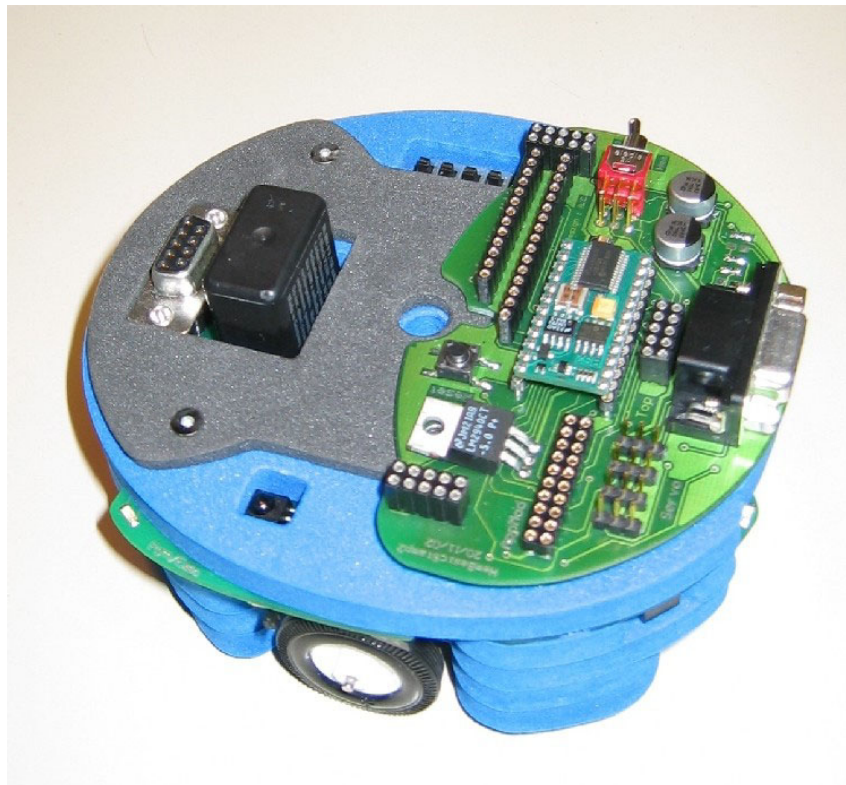
Marque Déposée :

Basic Stamp : Parallax, Inc.

Indications :

- Le contenu de ce manuel est susceptible de changer sans préavis. Une version à jour se trouve sur <http://www.hemisson.com>
- Tous les efforts ont été faits pour garantir l'exactitude de ce manuel. Si toutefois vous trouvez une erreur, merci de nous la communiquer.

Manuel d'utilisation du module HemBasicStamp pour Hemisson



Février 2002

Table des matières

| | |
|-------------------------------------------------|----|
| 1. Introduction | 4 |
| 2. Installation du module | 4 |
| 3. Vue d'ensemble..... | 5 |
| 3.1 Prise RS232..... | 5 |
| 3.2 Commutateur Hem/Serial | 6 |
| 3.3 Régulateur 5V..... | 6 |
| 3.5 Bouton Reset..... | 7 |
| 3.6 Connecteur AppMode..... | 7 |
| 3.7 Connecteur alimentation..... | 7 |
| 3.8 Connecteur I/O | 7 |
| 3.9 Connecteurs Servo | 7 |
| 4. Logiciel | 8 |
| 4.1 BASIC Stamp Editor | 8 |
| 4.1.1 Compiler un programme | 8 |
| 4.1.2 Charger un programme | 8 |
| 4.2 Mise à jour de Hemisson | 8 |
| 5. Commandes..... | 9 |
| 5.1 Charger un programme et l'exécuter | 9 |
| 5.2 Vitesse des moteurs | 9 |
| 5.3 Emettre un son | 10 |
| 5.4 Lecture des valeurs des interrupteurs..... | 10 |
| 5.5 Allumer/Eteindre les leds | 10 |
| 5.6 Lecture des valeurs de proximité | 11 |
| 5.7 Lecture des valeurs de luminosité | 12 |
| 5.8 Dernier octet reçu par le récepteur TV..... | 13 |
| 5.9 Lecture de la version de l'OS..... | 13 |
| 6. Modules d'extension standard Bstamp..... | 14 |
| 6.1 Connecteur standard AppMode..... | 14 |
| 6.2 Connecteurs I/O..... | 14 |
| 6.3 Connecteurs servo..... | 14 |
| 6.4 RS232..... | 14 |
| 7. Annexes | 15 |
| 7.1 Schéma électronique | 15 |
| 7.2 Connexions..... | 16 |

1. Introduction

Ce module HemBasicStamp a été conçu pour le robot Hemisson. Il permet de contrôler Hemisson à partir d'un environnement BASIC Stamp. Hemisson est donc utilisé comme base robotique.

Différents modules d'extension standard Bstamp peuvent être connectés sur ce module.

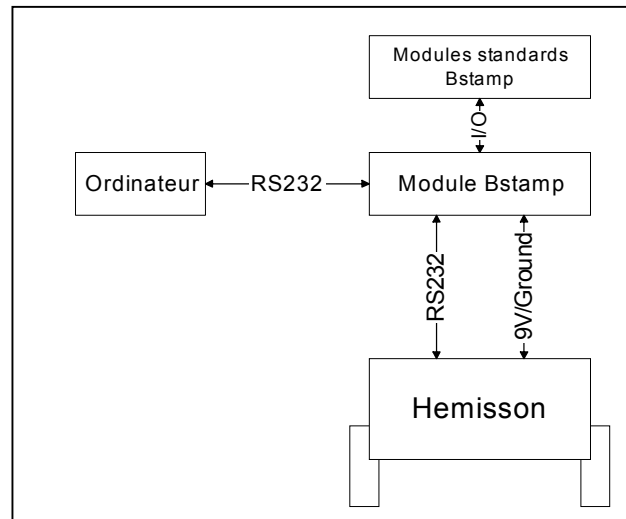


Fig. 1 Vue générale 1

2. Installation du module

Le module HemBasicStamp se connecte au robot Hemisson au moyen des 3 connecteurs d'extension. Il suffit d'insérer le module dans les connecteurs prévu à cet effet (Fig. 2 : 1,2,3).

Attention à ne pas tordre les pins du module et à bien tous les insérer dans les connecteurs.

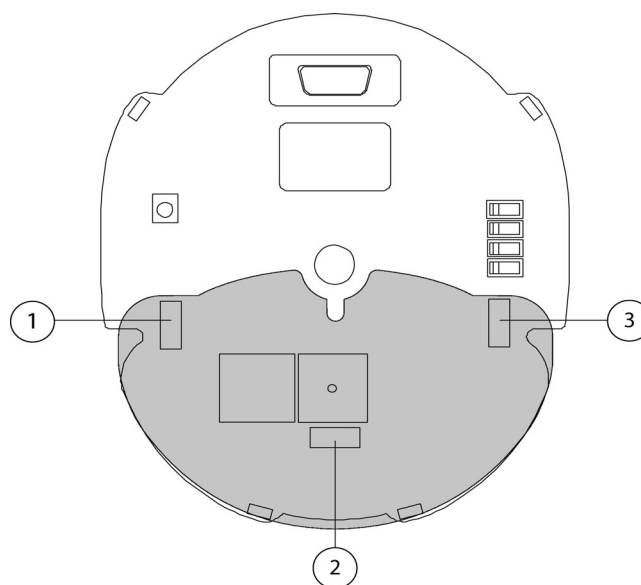


Fig. 2 Installation du module

3. Vue d'ensemble

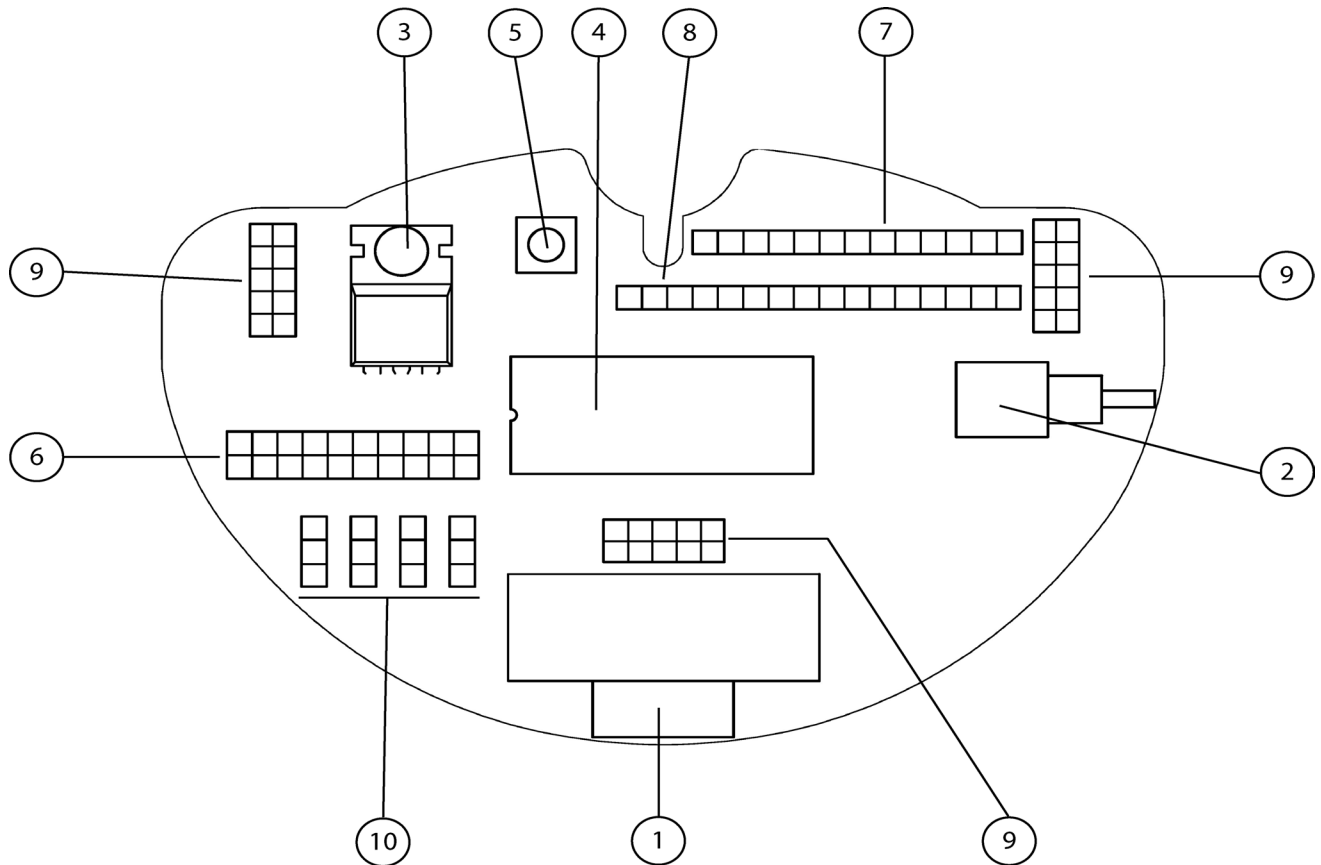


Fig. 3 Vue d'ensemble

1. Prise RS232
2. Commutateur Hem/Serial
3. Régulateur 5V
4. Emplacement pour microprocesseur Bstamp 24 pins
5. Bouton Reset
6. Connecteur AppMode
7. Connecteur alimentation
8. Connecteur I/O
9. Connecteurs Hemisson
10. Connecteurs Servo

3.1 Prise RS232

La prise RS232 permet de relier HemBasicStamp à un ordinateur afin de programmer le microprocesseur Bstamp. Pour cela votre ordinateur doit disposer d'un port série de type RS232 au format DB9. Vous pouvez utiliser le câble fournit avec le robot Hemisson.

Il existe des convertisseurs USB-Série si vous ne disposez pas d'un port série.

3.2 Commutateur Hem/Serial

Le commutateur permet de choisir avec qui communique le Bstamp : avec l'ordinateur ou Hemisson.

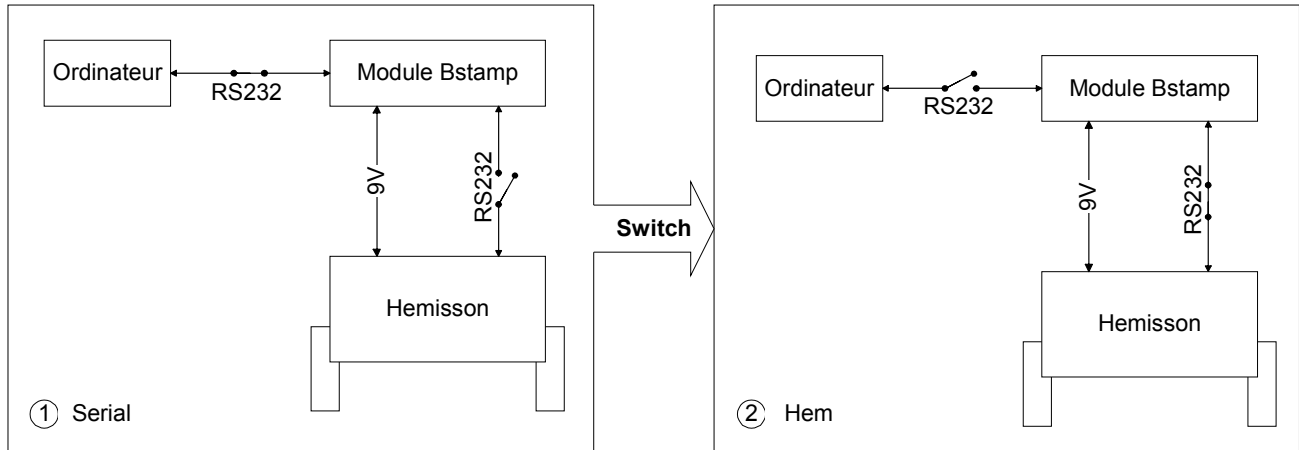


Fig. 4 Commutation Hem/Serial

Les Pins Sin(1) et Sout(2) sont utilisés dans les deux cas pour la communication entre l'ordinateur ou Hemisson. Ils ne sont donc pas disponibles pour d'autres utilisations.

1. Mode Serial

Ce mode permet de relier HemBasicStamp à l'ordinateur au moyen du port série. L'utilisateur peut ainsi programmer le microprocesseur Bstamp.

2. Mode Hem

Ce mode permet de contrôler le robot Hemisson. HemBasicStamp peut ainsi communiquer avec le microprocesseur d'Hemisson en utilisant les commandes sérielles prédéfinies. Pour plus d'informations sur les commandes d'Hemisson, se référer au manuel Hemisson, pages 23-25.

3.3 Régulateur 5V

Le régulateur de tension utilisé est un LM2940CT5. Il régule la tension de 9V provenant de la batterie d'Hemisson en une tension continue de 5 V.

3.4 Emplacement pour microprocesseur Bstamp 24 pins

Le logement prévu pour le microprocesseur Bstamp comporte 24 Pin. Il accepte les différentes versions du Bstamp (2, 2e, 2p, 2pe, 2sx, et ultérieures). Les exemples donnés dans ce manuel sont écrits pour le Bstamp 2. Le Bstamp 1 ne peut pas être connecté sur ce module! Les informations relatives à ces différents microprocesseurs sont disponibles sur le site www.parallax.com

3.5 Bouton Reset

Le bouton Reset permet de réinitialiser le Bstamp. Toutes les entrées/sorties passent à 0, ce qui réinitialise le programme à la première ligne.

3.6 Connecteur AppMode

Le connecteur AppMode est un connecteur standard femelle (20 Pin, pas : 2.54 mm) permettant de connecter divers modules supplémentaires. Il est connecté aux 16 I/O, au Gnd (masse), au Vdd (5V) et au Vin (9V).

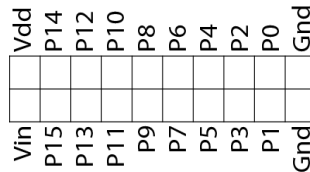


Fig. 5 Connecteur AppMode

3.7 Connecteur alimentation

Ce connecteur femelle comporte 13 pin (pas : 2.54 mm). Il fournit l'alimentation nécessaire pour des modules supplémentaires. 5 pins sont connectés au Gnd (5V), 3 au Gnd (masse) et 5 au Vin (9V) :



Fig. 6 Connecteur Alimentation

3.8 Connecteur I/O

Ce connecteur femelle comporte 16 pin (pas : 2.54 mm). Les 16 Pins sont connectés aux 16 pins I/O du Bstamp et permettent de brancher des modules supplémentaires :

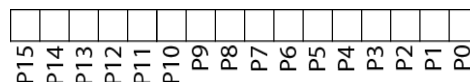


Fig. 7 Connecteur I/O

3.9 Connecteurs Servo

Les 4 connecteurs mâles (pas : 2.54 mm) permettent de connecter des servomoteurs. Chaque connecteur est relié à un Pin du Bstamp (12-15), à Vin (9V) et à Gnd (masse) :



Fig. 8 Connecteurs Servo

4. Logiciel

4.1 BASIC Stamp Editor

Le logiciel "BASIC Stamp Windows Editor" permet d'écrire du code en Pbasic et de le charger sur le Bstamp au moyen du câble RS232. Il fonctionne sous Windows (une version DOS existe aussi). Il peut être téléchargé gratuitement sur le site <http://www.parallax.com>

4.1.1 Compiler un programme

Ecrire votre code Pbasic dans un nouveau fichier puis cliquer sur le menu "Run" puis sur "Check syntax". L'éditeur indique si le programme contient des erreurs de syntaxe. Il existe une icône pour la commande "Check syntax" (Fig.22).

4.1.2 Charger un programme

Connecter le port série de HemBasicStamp à celui de l'ordinateur au moyen du câble RS232. Commuter HemBasicStamp sur "Serial", allumer Hemisson, cliquer sur le menu "Run", puis sur "Run". Le programme se charge alors dans le Bstamp. Il existe une icône pour la commande "Run" (Fig.22).

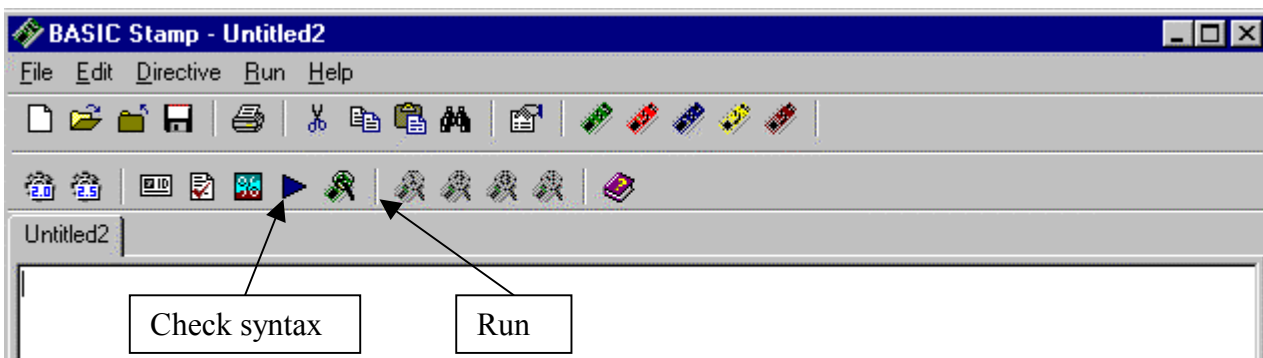


Fig. 9 Basic Stamp Editor

4.2 Mise à jour de Hemisson

Hemisson communique par défaut à une vitesse de 115'200 bauds via le port série. La communication avec HemBasicStamp doit se faire à des vitesses plus basses. Le Bstamp 2 accepte des vitesses de 243 à 50K bauds. Une vitesse plus haute que 2400 bauds peut poser des problèmes de synchronisation du Bstamp car celui-ci ne possède pas de buffer. Pour plus d'informations, se référer au manuel de référence Bstamp.

La communication s'effectue à 9600 bauds entre Hemisson et le Bstamp, donc une autre version du logiciel doit être chargée dans Hemisson au moyen du programme "Hemisson uploader", disponible sur le site www.hemisson.com avec sa documentation. Charger le fichier "main.HEX" dans Hemisson. Si le HemBasicStamp est connecté à Hemisson, mettre le commutateur en position "Serial" durant la mise à jour d'Hemisson.

5. Commandes

Un certain nombre de commandes permettent de contrôler le robot Hemisson et de connaître l'état de ses capteurs/interrupteurs. Toutes les commandes suivantes sont écrites pour le Bstamp 2, en Pbasic 2 mais compatible 2.5, et sont disponibles sur le site www.hemisson.com

5.1 Charger un programme et l'exécuter

1. Eteindre Hemisson (interrupteur sur Off)
2. Installer HemBasicStamp sur Hemisson
3. Commuter les interrupteurs de Hemisson en mode basique :



Fig.10 Mode basique

4. Mettre le commutateur du module HemBasicStamp sur "Serial"
5. Brancher le câble RS232 entre le port série de l'ordinateur et le module HemBasicStamp
6. Allumer Hemisson (interrupteur sur On)
7. Charger le programme dans le Bstamp au moyen du programme "Basic Stamp editor"
8. Mettre le commutateur sur "Hem"
9. Appuyer sur le bouton Reset de HemBasicStamp

5.2 Vitesse des moteurs

Donne des consignes de vitesses aux moteurs gauche et droit. Une valeur entre -9 et 9 doit être entrée pour les variables "gauche" et "droite". Une valeur égale à 0 correspondant à une vitesse nulle.

```
{$STAMP BS2}
'Assigner la vitesse des moteurs
gauche  VAR      WORD
droite  VAR      WORD

gauche=  5      'vitesse moteur gauche: [-9, 9], 0=arrêt
droite= -5      'vitesse moteur droite: [-9, 9], 0=arrêt

SEROUT 16, 84, ["D," , SDEC gauche, "," , SDEC droite, CR]
```

5.3 Emettre un son

Permet de produire un son continu avec le buzzer. Un '1' dans la variable "etat" enclenche le son et un '0' l'arrête.

```
{$STAMP BS2}
'Emettre un son

etat      VAR      BIT
etat=     1        '0:arret, 1:marche

SEROUT    16, 84, ["H," , DEC etat, CR]
```

5.4 Lecture des valeurs des interrupteurs

Permet de connaître la position des 4 interrupteurs. Un '0' correspond à un interrupteur ouvert et un '1' à un interrupteur fermé. Les valeurs sont stockées dans les variables "answ1-4".

```
{$STAMP BS2}
'Lecture des valeurs des interrupteurs

answ1     VAR      BIT
answ2     VAR      BIT
answ3     VAR      BIT
answ4     VAR      BIT

SEROUT    16, 84, ["I", CR]
SERIN     16, 84, [answ1,answ1,answ2,answ2,answ3,answ3,answ4,answ4]
```

5.5 Allumer/Eteindre les leds

Permet d'allumer ou d'éteindre les 4 leds. Une valeur de '0' correspond à une led éteinte et un '1' à une led allumée. Les valeurs sont stockées dans les variables "led1-4".

```
{$STAMP BS2}
'Allumer/Eteindre les leds (0:eteinte, 1:allumee)

led1      VAR      BIT
led2      VAR      BIT
led3      VAR      BIT
led4      VAR      BIT

led1 =    1        'On/Off
led2 =    1        'Pgm/Exec
led3 =    1        'Avant-gauche
led4 =    1        'Avant-droite

SEROUT    16, 84, ["L," , DEC led1, "," , DEC led2, "," , DEC led3, "," , DEC led4, CR]
```

5.6 Lecture des valeurs de proximité

Renvoie les valeurs de proximité de chaque capteur. Conformément à la courbe de réponse des capteurs Irs (manuel Hemisson, p.11) en mode actif, une valeur proche de zéro indique qu'il n'y a pas d'obstacles dans la direction du capteur, à l'inverse une valeur de 255 révèle la présence toute proche d'un obstacle. Il y a 3 fonctions différentes, pour les différents capteurs :

- Pour les capteurs avant, avant-droit et avant-gauche. Les valeurs sont stockées dans les variables "front ", "frontright", et "frontleft" :

```
{ $STAMP BS2}

'Lecture des valeurs de proximité
'front, frontright, frontleft

front          VAR          BYTE
frontright    VAR          BYTE
frontleft     VAR          BYTE

SEROUT        16, 84, ["P,0", CR]
SERIN         16, 84, [b10,b10,b10,b11,b12,b13,b13,b14,b15,b16,b16,b17,b18]

front =       (b10-48)*10+(b11-48)*10+(b12-48)
frontright = (b13-48)*10+(b14-48)*10+(b15-48)
frontleft =   (b16-48)*10+(b17-48)*10+(b18-48)
```

- Pour les capteurs droite, gauche et arrière. Les valeurs sont stockées dans les variables "right ", "left", et "rear" :

```
{ $STAMP BS2}

'Lecture des valeurs de proximité
'right, left, rear

right          VAR          BYTE
left           VAR          BYTE
rear           VAR          BYTE

SEROUT        16, 84, ["P,1", CR]
SERIN         16, 84, [b10,b10,b10,b11,b12,b13,b13,b14,b15,b16,b16,b17,b18]

right =       (b10-48)*10+(b11-48)*10+(b12-48)
left =        (b13-48)*10+(b14-48)*10+(b15-48)
rear =        (b16-48)*10+(b17-48)*10+(b18-48)
```

- Pour les capteurs sol-droite et sol-gauche. Les valeurs sont stockées dans les variables "groundright" et "groundleft" :

```
{ $STAMP BS2}

'Lecture des valeurs de proximité
'groundright, groundleft

groundright   VAR          BYTE
groundleft    VAR          BYTE

SEROUT        16, 84, ["P,2", CR]
SERIN         16, 84, [b10,b10,b10,b11,b12,b13,b13,b14,b15]

groundright = (b10-48)*10+(b11-48)*10+(b12-48)
groundleft =  (b13-48)*10+(b14-48)*10+(b15-48)
```

5.7 Lecture des valeurs de luminosité

Renvoie les valeurs de luminosité de chaque capteur. Une valeur élevée (255) indique qu'aucune lumière infrarouge n'est détectée dans l'axe du capteur. . Il y a 3 fonctions différentes, pour les différents capteurs :

- Pour les capteurs avant, avant-droit et avant-gauche. Les valeurs sont stockées dans les variables "front ", "frontright", et "frontleft" :

```
{ $STAMP BS2}

'Lecture des valeurs de luminosité des capteurs
'front, frontright, frontleft

front          VAR          BYTE
frontright    VAR          BYTE
frontleft     VAR          BYTE

SEROUT        16, 84, ["M,0", CR]
SERIN         16, 84, [b10,b10,b10,b11,b12,b13,b13,b14,b15,b16,b16,b17,b18]

front =       (b10-48)*10+(b11-48)*10+(b12-48)
frontright = (b13-48)*10+(b14-48)*10+(b15-48)
frontleft =   (b16-48)*10+(b17-48)*10+(b18-48)
```

- Pour les capteurs droite, gauche et arrière. Les valeurs sont stockées dans les variables "right ", "left", et "rear" :

```
{ $STAMP BS2}

'Lecture des valeurs de luminosité des capteurs
'right, left, rear

right         VAR          BYTE
left          VAR          BYTE
rear          VAR          BYTE

SEROUT        16, 84, ["M,1", CR]
SERIN         16, 84, [b10,b10,b10,b11,b12,b13,b13,b14,b15,b16,b16,b17,b18]

right =       (b10-48)*10+(b11-48)*10+(b12-48)
left =        (b13-48)*10+(b14-48)*10+(b15-48)
rear =        (b16-48)*10+(b17-48)*10+(b18-48)
```

- Pour les capteurs sol-droite et sol-gauche. Les valeurs sont stockées dans les variables "groundright" et "groundleft" :

```
{ $STAMP BS2}

'Lecture des valeurs de luminosité des capteurs
'groundright, groundleft

groundright  VAR          BYTE
groundleft   VAR          BYTE

SEROUT        16, 84, ["M,2", CR]
SERIN         16, 84, [b10,b10,b10,b11,b12,b13,b13,b14,b15]

groundright = (b10-48)*10+(b11-48)*10+(b12-48)
groundleft =  (b13-48)*10+(b14-48)*10+(b15-48)
```


6. Modules d'extension standard Bstamp

Différents modules compatibles Bstamp peuvent être connectés à HemBasicstamp. Ils peuvent être connectés de différentes façons :

6.1 Connecteur standard AppMode

Plusieurs modules standard peuvent directement se connectés à ce connecteur. Ce connecteur comprend l'alimentation ainsi que les pin I/O. Plusieurs modules peuvent être connectés à ce connecteur, à condition qu'ils emploient des I/O différentes. La documentation relative à ces modules se trouve sur le site www.parallax.com.

Exemples: Led Terminal, Audio Amplifier, Sound module, Line follower, Modem, Compass

6.2 Connecteurs I/O

Différents modules peuvent être connectés au connecteur I/O au moyen de câbles standard. Ils seront alimentés par le connecteur alimentation, soit en 9V ou en 5V.

Exemples: Wireless RF, affichage LCD, Infrared distance sensor, Speech board, Serial servo controller

6.3 Connecteurs servo

Des moteurs servo peuvent directement être connectés à ces connecteurs au moyen d'un câble à embout femelle 3 pin (2.54 mm). Ces connecteurs comprennent une alimentation 9V. Il y a 4 connecteurs donc de la place pour 4 moteurs. Les I/O employées sont les 12-15 donc attention à ne pas les employer en même temps avec le connecteur AppMode ou les connecteurs I/O.

6.4 RS232

HemBasicStamp utilise les pins Sin/Sout pour communiquer avec Hemisson. Il faut donc utiliser directement des I/O pour envoyer et recevoir de l'information en utilisant RS232 par les commandes Pbasic SERIN/SEROUT.

Exemples: cf. Connecteurs I/O.

7. Annexes

7.1 Schéma électronique

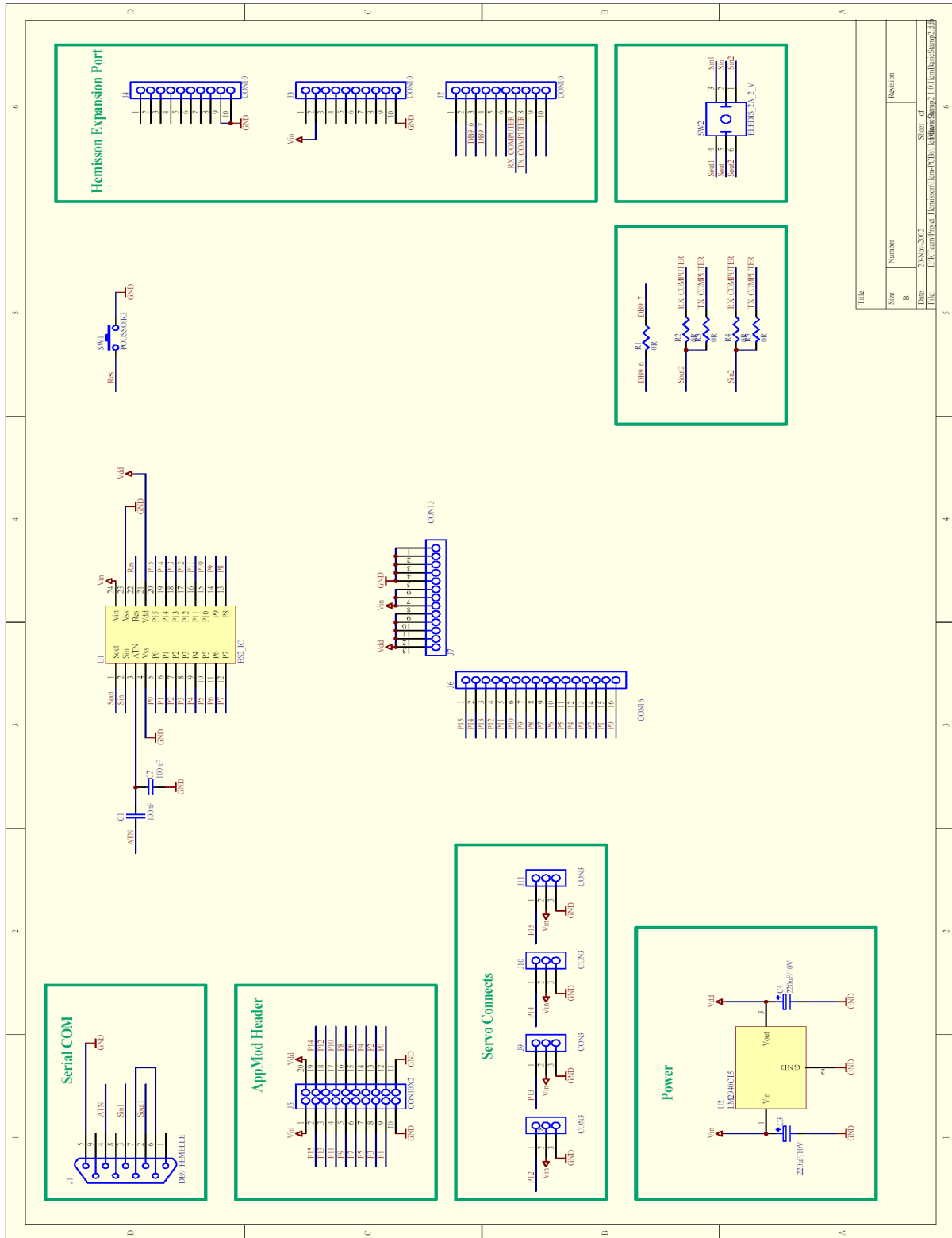


Fig. 11 Schéma électronique

7.2 Connexions

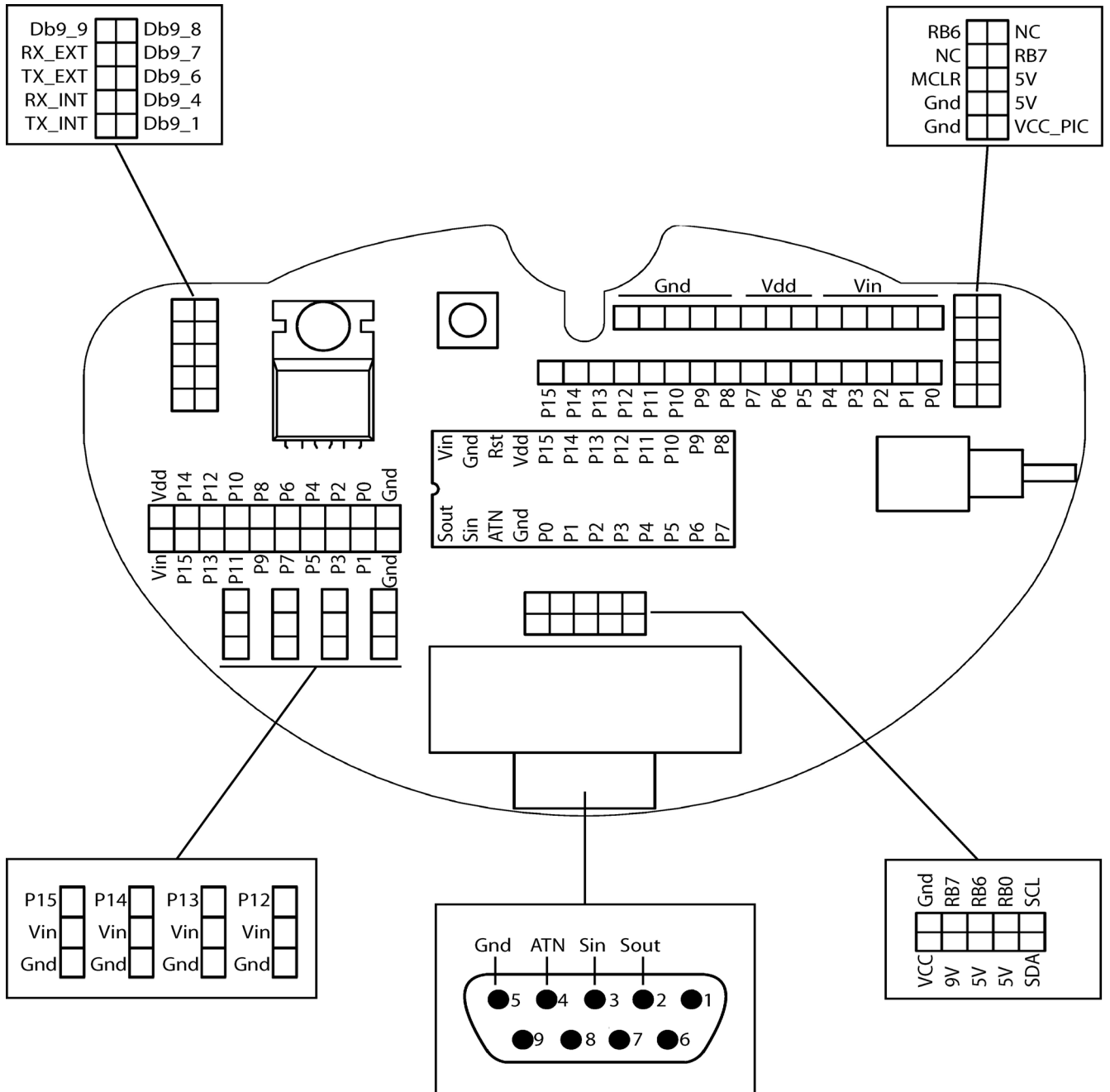


Fig. 12 Connexions