

J.D. Nicoud, Mouette 5
CH-1092 Belmont, Switzerland
Tel +41 21 728-6156, Fax 728-6157
Email info@didel.com, www.didel.com

DIDEL

Spécialistes en micro-robotique

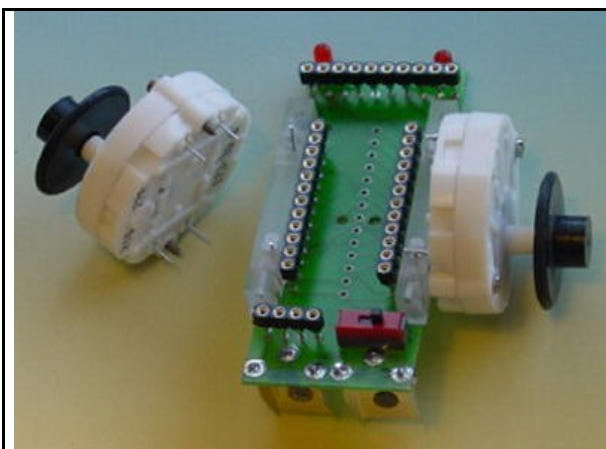
Swibot/Stamp

Le BASIC STAMP peut être une étape intéressante pour l'amateur qui désire découvrir le monde magique des microcontrôleurs. Il est très simple à programmer, il suffit de connaître les rudiments du BASIC.

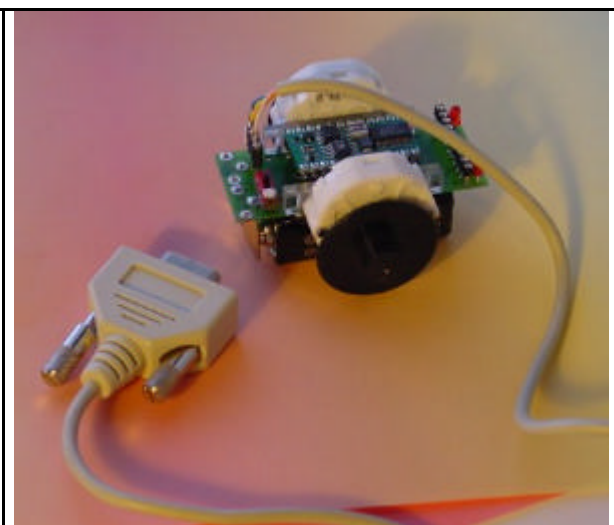
Les BASIC STAMP BS1 et BS2 de Parallax (<http://www.parallax.com>) sont des petits circuits imprimés qui contiennent un microcontrôleur PIC, qui exécute un programme interprété préparé en BASIC sur un PC. Interprété veut dire que le programme que vous allez écrire puis charger dans votre STAMP va être traduit par celui-ci au moment de son exécution. Ceci ralentit considérablement l'exécution, mais pour beaucoup d'application le traitement de 2000 instructions par seconde est suffisamment performant !

Le BASIC STAMP1 est limité à une centaine d'instructions. Il permet d'allumer des lampes et interagir avec des capteurs simples. C'est plus amusant est de commander un robot mobile, mais le bricolage d'un moteur, réducteur, alimentation est cher est nécessite de bonnes connaissances mécaniques et électroniques, Deux robots commerciaux sont compatibles avec le BASIC STAMP (<http://www.parallaxinc.com/stamps/growbot.htm> and <http://cantronics.rzsoft.com/pocketbo.htm>), mais ils ne sont pas aussi simples, avantageux et efficaces que le robot Swibot, qui existe aussi en version avec un Pic 16F84 directement programmée en assembleur CALM avec l'éditeur Smile NG, et une syntaxe d'assembleur très lisible. Pour celui qui ne possède pas déjà un BS1 ou BS2, ce Swibot est une solution économique avec des perspectives de programmation infinies, et des modules d'extension aussi passionnants qu'une petite caméra linéaire.

Le SWITEC/STAMP est adorablement petit et il consomme très peu sur sa batterie 9V. En kit, c'est un plaisir de le monter: il suffit d'un fer à souder à panne fine et d'un petit tournevis.



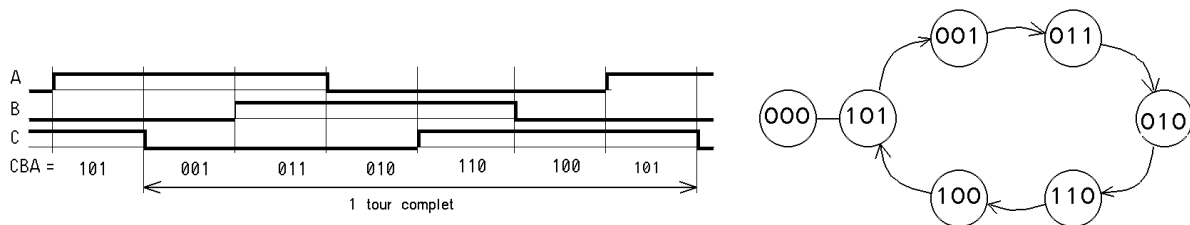
Le Kit facile à monter



Prêt à être chargé en série par le PC

Le moteur pas à pas comporte un réducteur 180. Pour chaque tour de moteur, qui nécessite un séquençement sur 6 phases, le robot avance de 0.1mm (v). La vitesse maximale est de 1ms par phase. Les phases des 2 moteurs sont commandées directement par les sorties du PIC, car le courant est de 20 mA seulement. Ceci utilise 6 sorties. Les deux sorties restantes du port permettent d'allumer deux diodes lumineuses. Avec le BS2 en boîtier DIL, il y a 7 sorties supplémentaires disponibles sur un connecteur d'extension. Un circuit en « piggy back » sur ce connecteur vous permettra d'ajouter vos capteurs.

Pour faire tourner un moteur de notre robot, nous devons envoyer une suite de 0 et de 1 sur ses 3 phases, selon le diagramme des temps ci-dessous. Un inversant la séquence, le moteur tourne dans l'autre sens. En mettant des 0 sur toutes les phases, le moteur ne consomme plus de courant, mais il est en roue libre. On peut l'arrêter sur n'importe quelle phase, mais nous avons choisi ici de faire des tours complet et de revenir toujours sur la même phase avant de change de sens. Il faut veiller que les phases se suivent dans l'ordre direct ou inverse, et durent toutes 1 ms au moins. Le moteur est assez tolérant, et il peut tourner à vide sur des séquences erronées, mais il n'a pas de puissance.



Nous voyons des colonnes numérotées de 7 à 0 correspondant aux 8 E/S de notre STAMP. Les colonnes 0 à 2 correspondent au moteur A et 3 à 5 au moteur B.

Le codage se fait en binaire, avec le préfixe x' On pourrait convertir le binaire en décimal, en tenant compte de poids 128, 64, 32, 16, 8, 4, 2, 1 des bits de gauche à droite, mais c'est plus

Les séquences pour faire avancer les deux moteurs, pour reculer, pour tourner sur place sont les suivantes:

	Dio des	Moteur B	Moteur A	Décimal
	7 6	5 4 3	2 1 0	
Avancer		0 0 1	0 0 1	9
		0 1 1	0 1 1	27
		0 1 0	0 1 0	18
		1 1 0	1 1 0	54
		1 0 0	1 0 0	36
		1 0 1	1 0 1	45
Reculer		1 0 0	1 0 0	36
		1 1 0	1 1 0	54
		0 1 0	0 1 0	18
		0 1 1	0 1 1	27
		0 0 1	0 0 1	9
		1 0 1	1 0 1	45

Tourner surplace	0	0	1	1	0	0	12
	0	1	1	1	1	0	30
	0	1	0	0	1	0	18
	1	1	0	0	1	1	51
	1	0	0	0	0	1	33
	1	0	1	1	0	1	45

Ci-dessous un exemple simple qui fait décrire un rectangle au robot, il peut par exemple être posé sur un livre et en suivre le bord. Les boucles FOR/NEXT devront être ajuster.

```
' Robot
Dir = %11111111 ' tout en sorties
Symbol P = 2 ' Déclare la constante P, qui permet de ralentir le STAMP

Pins = %00101101 ' Moteur en phase initiale, diodes éteintes

debut: ' Petite pause pour laisser le temps de mettre en marche et de
poser le robot.

for b4 = 1 to 240
Pins = x'11000000
'-- on pourrait clignoter pour mieux montrer que cela est bien parti
next
'Les lampes sont éteintes mais on pourrait les activer ou désactiver dans chaque boucle "for"
for b2 = 0 to 240 'avant coté long
Pins = x'00001001
Pause P ' attend 2 cycles pour ralentir
Pins = x'00011011
Pause P
Pins = x'00010010
Pause P
Pins = x'00110110
Pause P
Pins = x'00100100
Pause P
Pins = x'00101101
Pause P
next

for b3 = 0 to 79 'quart de¼ tour
Pins = x'00001100
Pause P
Pins = x'00011110
Pause P
Pins = x'00010010
Pause P
Pins = x'00110011
Pause P
Pins = x'00100001
```

```

    Pause P
    Pins = x'00101101
    Pause P
next

for b2 = 0 to 120      'avant coté court.
    Pins = x'00001001
    Pause P
    Pins = x'00011011
    Pause P
    Pins = x'00010010
    Pause P
    Pins = x'00110110
    Pause P
    Pins = x'00100100
    Pause P
    Pins = x'00101101
    Pause P
next

for b3 = 0 to 78      'quart de tour
    Pins = x'00001100
    Pause P
    Pins = x'00011110
    Pause P
    Pins = x'00010010
    Pause P
    Pins = x'00110011
    Pause P
    Pins = x'00100001
    Pause P
    Pins = x'00101101
    Pause P
next

```

Goto debut

End

Si le robot ne parvient pas à faire exactement 90 °, il faut corriger le nombre de pas.

Programme écrit par J.C. Sulka, F-18203 St-Amand MTD, 1999