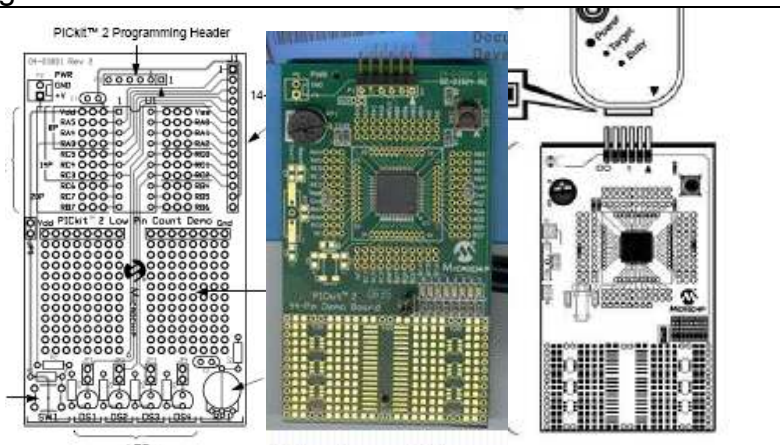


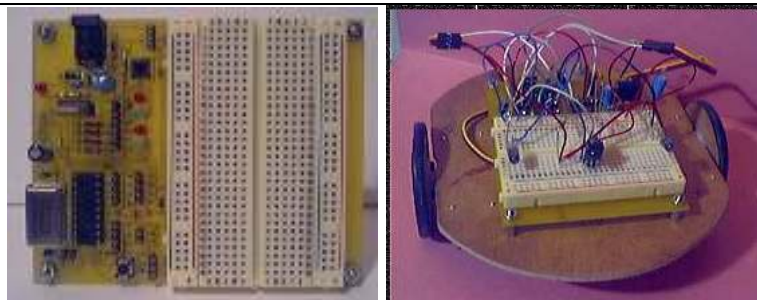
Modules flexibles pour le développements d'applications avec les microcontrôleurs PIC et AVR

Quelles sont les solutions pour développer une application avec un microcontrôleur et des interfaces variés ? Seules les application qui ont des contraintes de temps réel nous intéressent. Jouer avec un affichage LCD ou un réseau I²C/Can nécessite d'autres outils.

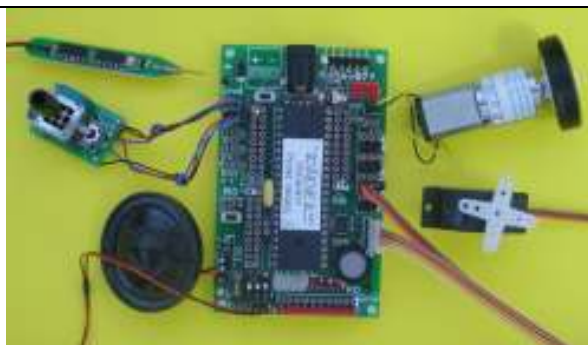
Microchip et d'autres proposent des cartes pour se familiariser avec la programmation. Elles comportent une zone universelle qui convient pour câbler les circuits annexes. Mais alors, la carte n'est plus utilisable pour le projet suivant.



Une carte d'expérimentation a l'avantage d'accepter une grande variété de composants, mais la fiabilité est mauvaise et le montage doit être refait si on veut redévelopper du soft ou faire des modifs qq mois plus tard. Avec peu de composants, on ne voit plus rien !



La carte Dev877 de Didel est optimisée pour un schéma avec des pot, moteurs, servos. Elle a des connecteurs pour câbler ce que l'on veut, mais elle est trop compacte pour expérimenter.

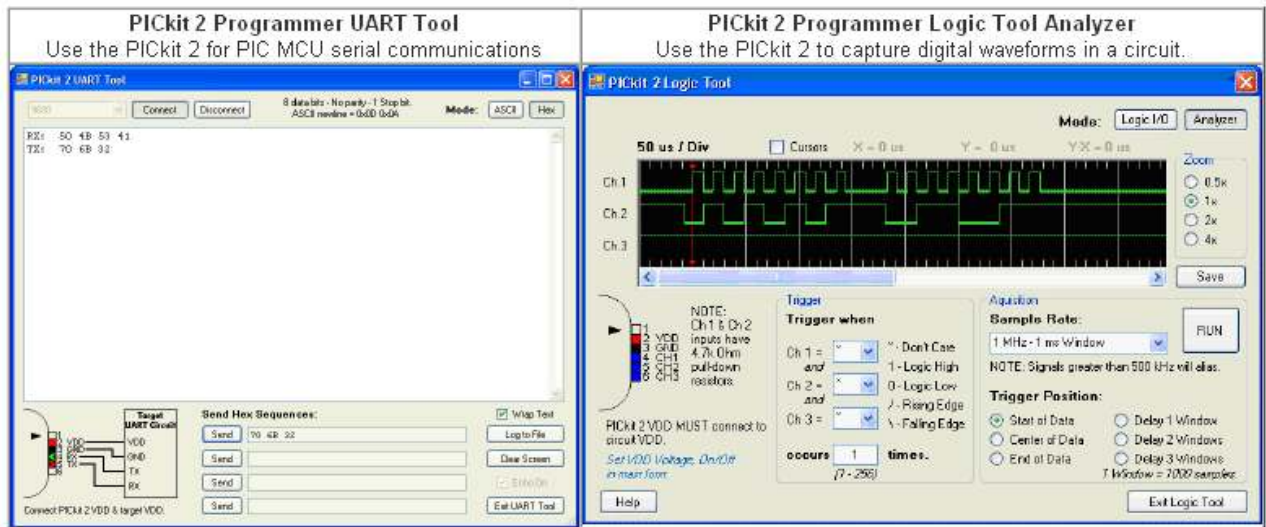


Les **Microdules** sont un ensemble de cartes pour faciliter le développements d'application avec des microcontrôleurs de toutes taille. Un choix de cartes compatibles avec les processeurs les plus courants est complété par des modules 8 bits se connectés sur les ports et adaptés pour mettre au point une application donné. Quant le schéma et le logiciel est testé le circuit imprimé définitif (ou Veroboard compact) est réalisé et connecté à l'application.

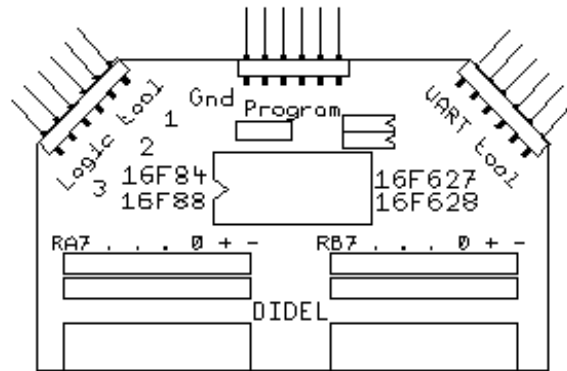
Les cartes processeurs n'ont que des connecteurs et sont prévues pour être connectées à un programmeur et à un ensemble de petites cartes interfaces, câblées en fonction de l'application.

Premier catalogue Microdude (projet juin 08)

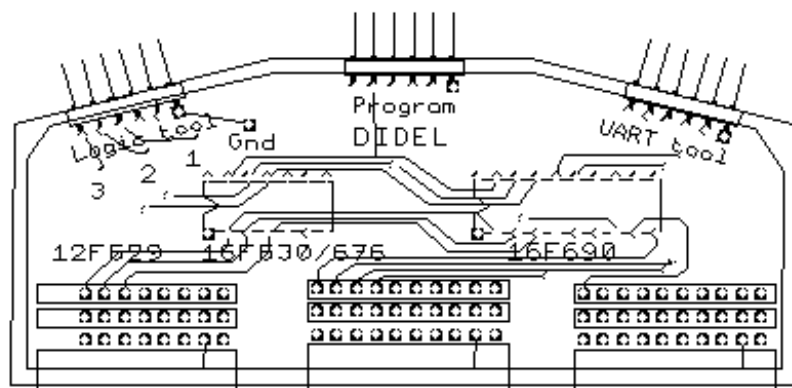
Les cartes compatibles avec les processeurs de Microchip sont optimisées pour le programmeur PicKit2 et ses deux fonctionnalités complémentaires, uart et analyseur logique. On peut déplacer le PicKit2 ou en avoir plusieurs.



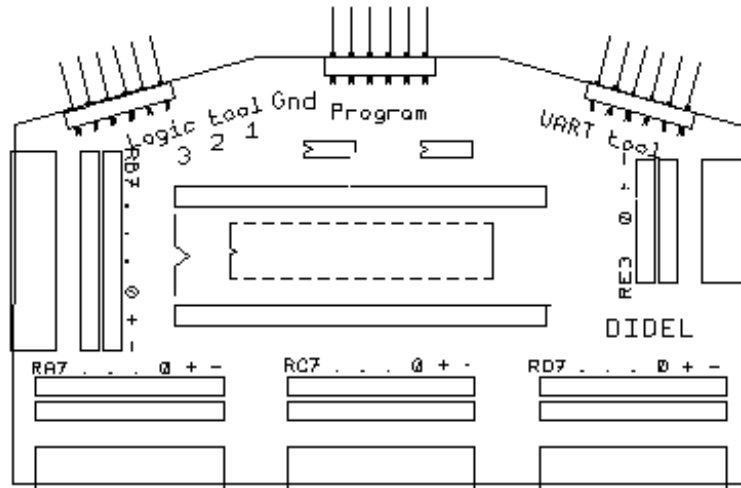
Carte pour F84/etc
(Ports A et B).



Carte pour 12F505
12F629/675 etc
(Port A 6 bits),
16F630/676/etc
(Ports A et C 6 bits),
16F690/etc
(Ports A et B et C 5-8 bits).



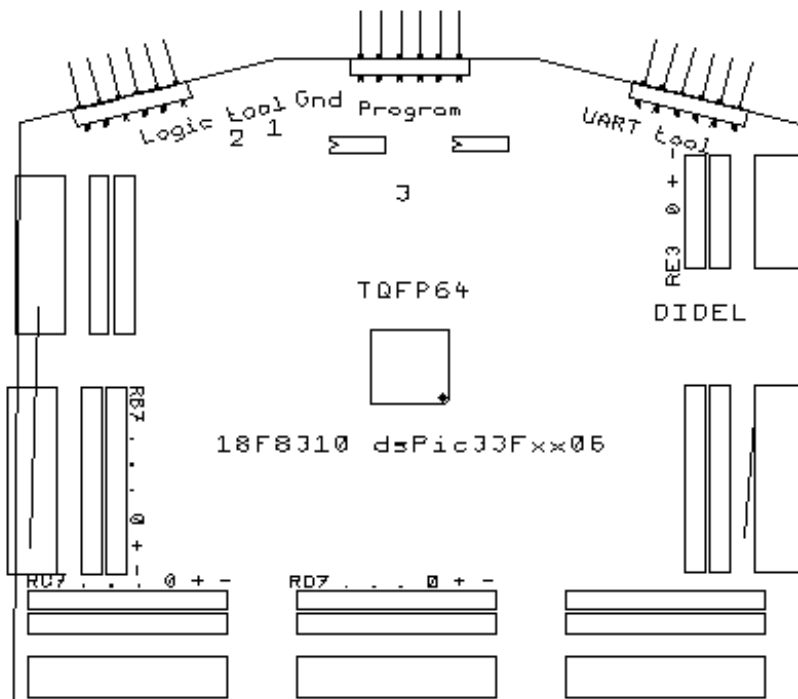
Carte pour 877/etc
Ports A B C D E



Carte pour processeurs
64 pattes SMD (Ports A
B C D E F G)

La carte pour
processeur 80 pattes
est plus longue (2 ports
de plus).

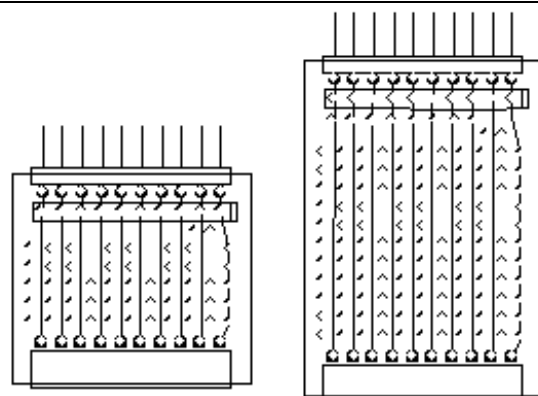
Ajouter un connecteur
ICD2 ?

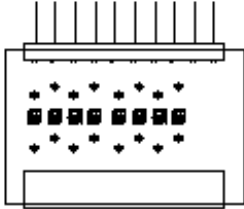


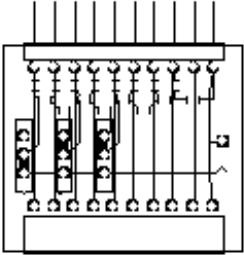
Pour des AVR,
MSP430 ou autre, il
suffit de changer le
routage du PCB

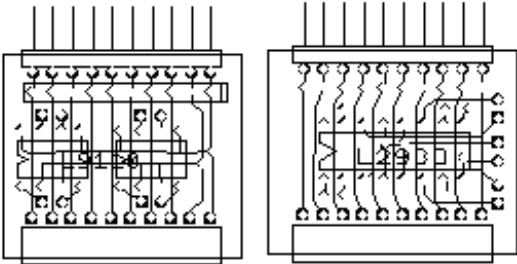
Carte universelle (genre Veroboard) pour
câblage des circuits branchés sur un port.

Sur la face dessus, les connecteurs du
haut et du bas sont reliés, sans toucher
les trous. Solder mask pour éviter un
court-circuit avec un fil non isolé.
Sur la face dessous, faut-il des liaisons
horizontales ou verticales (faciles à
couper, 0.4mm)

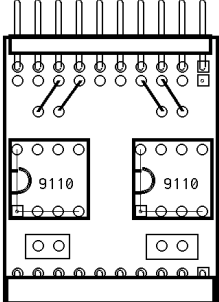
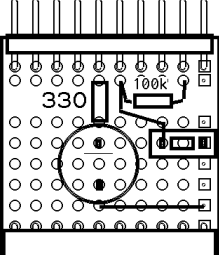
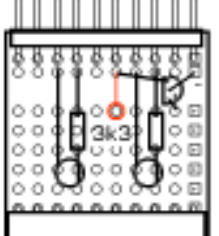


<p>Carte d'affichage Les Leds affichent l'état des 8 lignes : état 0 – rouge, état1-vert, état en l'air – orange. La consommation est de 0.03mA par ligne (résistance série de 10 kOhm)</p>	
--	---

<p>Conv D/A pour aide mise au point Trois commutateurs permettent des visualiser sur oscillo un compteur ou pointeur 4 bits, 6 bits ou 8 bits.</p>	
---	---

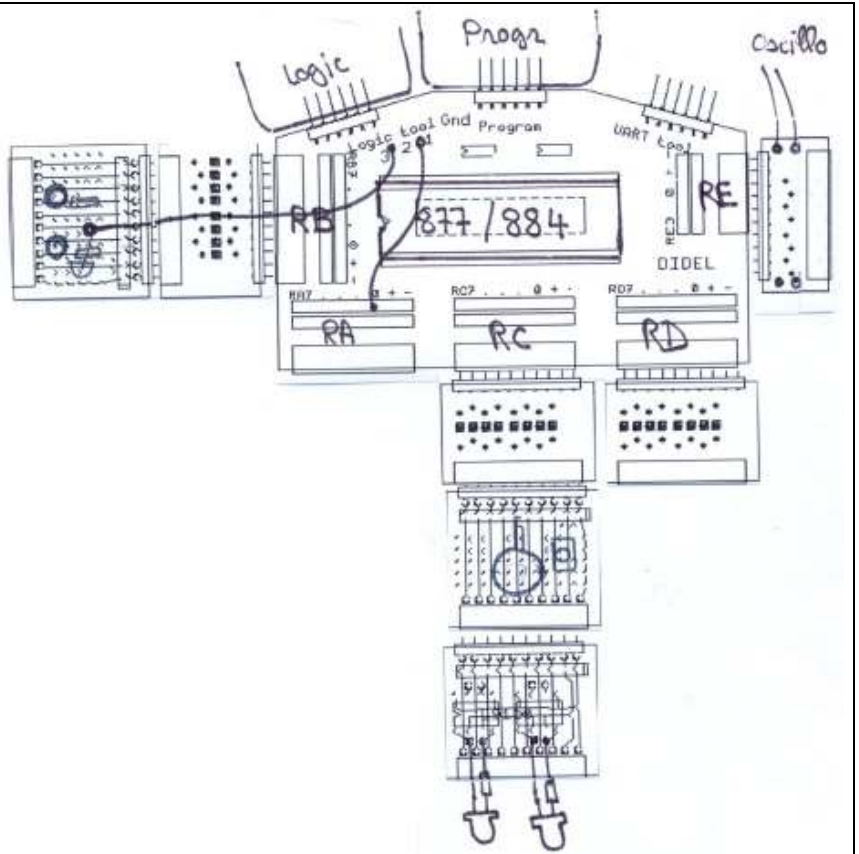
<p>Cartes spécialisées 2x 9110 ampli moteur 0.5A 1x L293D ampli moteur 12V-1A Autres cartes ?? Commande de moteur pas-à-pas Camera linéaire</p>	
--	---

--	--

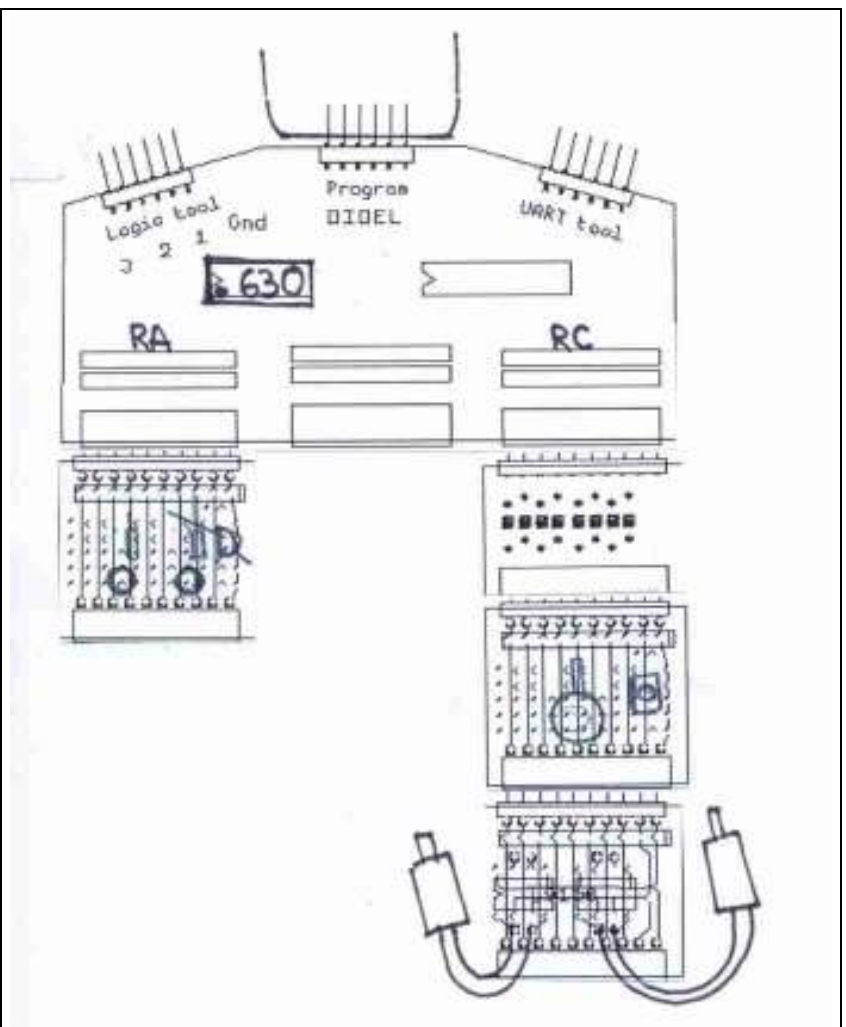
<p>Exemple d'application Pour tester le Bimo on doit câbler RC0 RC1 moteur gauche RC4 RC5 moteur droit Le module L9110 est câblé pour que les entrées soient sur les bons bits. Les moteurs ou des leds bicolores se branchent sur les connecteurs de sortie</p>	
<p>RC2 poussoir RC3 haut-parleur Un module vero est facile à câbler et reste clair Il sera cascadié avec le module moteur puisque c'est sur le même port.</p>	
<p>RA2 entrée module IR (récepteur infrarouge) RA0 RA1 LED bicolore RA4 RA5 LED bicolore (RA3 programmation) On câblera une douille 0,5mm pour aller sur l'analyseur logique du Pickit2, ou un oscilloscope, au moment de la mise au point du décodage de la réception infrarouge.</p>	

Pour le développement, on utilise un 16F877 ou 16F884 avec le PortB à la place du portA du 630 du schéma (sur le 877 RA4 est open collector).

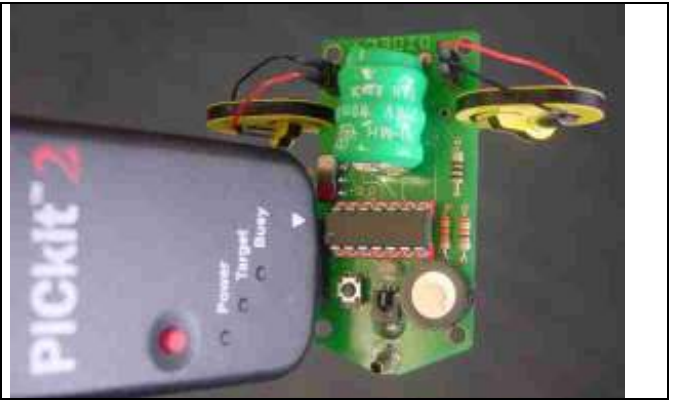
Les Ports A, D et E sont libres pour la mise au point. On utilise RA0, RA1 en sortie comme signaux de synchronisation pour l'analyseur logique ou l'oscillo. On utilise le PortE avec un convertisseur D/A pour le suivi des machines d'état, le portD avec ses LEDs ou un convertisseur D/A pour observer des données, des compteurs.



La compatibilité avec le 630 (.If dans le programme) est testée en déplaçant les modules et en réassemblant. La mise au point est terminée, mais on peut devoir régler des paramètres et l'analyseur logique peut encore être utile.



Le transport du logiciel sur le Bimo n'a pas de raison de poser problème. Pas de fil temporaire à souder pour aider à la mise au point !



Les circuits imprimés pourront se vendre 1-3 francs et les barrettes 0.20 à 0.40 cts pour 10 contacts. Donc pas besoin de les économiser. Ils restent câblés et disponibles pour une amélioration de soft ultérieure.

jdn 080609