Notes d'applications
Volume #A

"Everything for Embedded Control"

COMFILE
TECHNOLOGY

Copyrights et appellations commerciales

Toutes les marques, les procédés et les références des produits cités dans ce document appartiennent à leur propriétaire et Fabricant respectif. All brand names and trademarks are the property of their respective owners - Other trademarks mentioned are registered trademarks of their respective holders.

Informations techniques

Les notes d'applications décrites dans ce document ont été conçues avec la plus grande attention. Tous les efforts ont été mis en oeuvre pour éviter les anomalies. Toutefois, nous ne pouvons garantir que ce document soit à 100% exempt de toute erreur. Les informations présentes dans ces notes d'applications sont strictement données à titre indicatif. Les caractéristiques et les résultats obtenus par ces notes d'applications peuvent changer à tout moment sans aucun préavis.

Limitation de responsabilité

En aucun cas LEXTRONIC ne pourra être tenu responsable de dommages quels qu'ils soient (intégrant, mais sans limitation, les dommages pour perte de bénéfice commercial, interruption d'exploitation commerciale, perte d'informations et de données à caractère commercial ou de toute autre perte financière) provenant de l'utilisation ou de l'incapacité à pouvoir utiliser les notes d'applications décrites dans ce document, même si LEXTRONIC a été informé de la possibilité de tels dommages.

Ces notes d'applications sont uniquement destinées à être utilisées telles quelles dans le cadre d'un apprentissage à la programmation des modules "CUBLOC™". LEXTRONIC ne donne aucune garantie de fonctionnement de ces notes d'applications si vous utilisez celles-ci au sein d'une autre application. A ce titre, ces notes d'applications ne sont pas conçues, ni destinées, ni autorisées pour être utilisées au sein d'applications militaires, ni au sein d'applications à caractère médical ou d'alerte incendie, ni au sein d'applications pour ascenseurs ou commande de feux d'artifices, ni au sein d'applications sur machine outils ou d'applications embarquées dans des véhicules (automobiles, camions, bateaux, scooters, motos, kart, scooters des mers, avions, hélicoptères, ULM, etc...), ni au sein d'applications embarquées sur des maquettes volantes de modèles réduits (type avions, hélicoptères, planeurs, etc...).

De manière générale, ces notes d'applications ne sont pas conçues, ni destinées, ni autorisées pour expérimenter, développer ou être intégrées au sein d'applications dans lesquelles une défaillance des modules "CUBLOC™" pourrait créer une situation dangereuse pouvant entraîner des pertes financières, des dégâts matériels, des blessures corporelles ou la mort de personnes ou d'animaux. Si vous utilisez ces notes d'applications associées aux modules "CUBLOC™" ainsi que leurs platines et modules optionnels associés volontairement ou involontairement pour de telles applications non autorisées, vous vous engagez à soustraire le Fabricant et LEXTRONIC de toute responsabilité et de toute demande de dédommagement.

L'exploitation de ces notes d'applications nécessite que l'utilisateur respecte également toutes les précautions d'utilisations relatives à la mise en œuvre des modules "CUBLOC™" (lesquelles sont détaillées dans la documentation de ces derniers).

Liste des notes d'applications par ordre chronologique de parution

#1 – Bargraph « LCD » de précision 6
#2 – Lecture de 10 boutons-poussoirs via une seule entrée du CUBLOC™ 11
#3 – Gestion d'un capteur « FSR » 15
#4 – Gestion d'un clavier matricé 19
#5 – Gestion d'un télémètre ultrason « MSU08 » 22
#6 – Gestion d'une boussole électronique « CMP03 » 25
#7 – Gestion d'un afficheur à dalle tactile « ezDISPLAY » 28
#8 – Gestion d'un module synthétiseur vocal « SP03 » 34
#9 – Gestion d'un capteur thermique « MTP81 ». 38
#10 – Mini jeu vidéo « CUBLOC ATTACK » 41
#11 – Mise en œuvre d'un modem radio « TDL2A-433-9 » 49
#12 – Configuration d'un modem radio « TDL2A-433-9 » 54
#13 – Interfaçage avec un module de gestion de clavier de PC « ezKEY » 59
#14 – Interfaçage avec un module de lecture RFID « UM005 » 63
#15 – Interfaçage avec un module de gestion de servomoteurs « SSC03A » 68
#16 – Interfaçage avec modules de gestion de moteurs 'cc' « MCM1/MCM2 » 71
#17 – Interfaçage avec un module de gestion de souris « ezMOUSE » 75
#18 – Interfaçage avec module de gestion de moteurs 'cc' « MD-22 » 80
#19 – Expérimentations avec différents types de capteurs 84
#20 – Pilotage de servomoteurs via les sorties « PWM » 87
#21 – Gestion des télémètres ultrason « MSU10 » et « MSU235 » 91
#22 – Interfaçage avec un module de gestion de servomoteurs « SMCPRO » 95
#23 – Interfaçage avec un module horloge temps réel « RTCBoard » à base de DS1302 ... 98
#24 – Gestion d'un afficheur LCD alphanumérique « Demmel products » 102
#25 – Gestion d'une base robotique « Rogue Blue » 107
#26 – Gestion d'une base robotique type « Araignée » 111
#27 – Gestion d'un afficheur graphique couleur « Micro-LCD » 115
#28 – Conversion d'un afficheur LCD alphanumérique en afficheur LCD graphique 121
#29 – Gestion d'une matrice à Leds à commande série « EFN » 126
#30 – Interfaçage avec un accéléromètre 2 axes « Acell » 128

Liste par "thèmes"

Gestion de modules d'affichages
#1 – Bargraph « LCD » de précision 6
#7 – Gestion d'un afficheur à dalle tactile « ezDISPLAY »..... 28
#24 – Gestion d'un afficheur LCD alphanumérique « Demmel products »..... 102
#27 – Gestion d'un afficheur graphique couleur « Micro-LCD »..... 115
#28 – Conversion d'un afficheur LCD alphanumérique en afficheur LCD graphique..... 121
#29 – Gestion d'une matrice à Leds à commande série « EFN » 126

Interfaçage avec capteurs
#3 – Gestion d'un capteur « FSR » 15
#5 – Gestion d'un télémètre ultrason « MSU08 » 22
#6 – Gestion d'une boussole électronique « CMP03 » 25
#9 – Gestion d'un capteur thermique « MTP81 »..... 38
#19 – Expérimentations avec différents types de capteurs..... 84
#21 – Gestion des télémètres ultrason « MSU10 » et « MSU235 » 91
#30 – Interfaçage avec un accéléromètre 2 axes « Acell » 128

Interfaçage avec périphériques de saisie
#2 – Lecture de 10 boutons-poussoirs via une seule entrée du CUBLOC™ 11
#4 – Gestion d'un clavier matricé 19
#13 – Interfaçage avec un module de gestion de clavier de PC « ezKEY »..... 59
#17 – Interfaçage avec un module de gestion de souris « ezMOUSE »..... 75

Interfaçage avec modules de restitution sonores
#8 – Gestion d'un module synthétiseur vocal « SP03 ».....34

Interfaçage avec modules de communications
#11 – Mise en œuvre d'un modem radio « TDL2A-433-9 »..... 49
#12 – Configuration d'un modem radio « TDL2A-433-9 »..... 54

Gestion de moteurs
#15 – Interfaçage avec un module de gestion de servomoteurs « SSC03A »..... 68
#16 – Interfaçage avec modules de gestion de moteurs 'cc' « MCM1/MCM2 »..... 71
#18 – Interfaçage avec module de gestion de moteurs 'cc' « MD-22 »..... 80
#20 – Pilotage de servomoteurs via les sorties « PWM »..... 87
#22 – Interfaçage avec un module de gestion de servomoteurs « SMCPRO »..... 95

Applications diverses

#10 – Mini jeu vidéo « CUBLOC ATTACK »…… **41**
#14 – Interfaçage avec un module de lecture RFID « UM005 »…… **63**
#23 – Interfaçage avec un module horloge temps réel « RTCBoard » à base de DS1302…. **98**

Robotique "ludique"

#25 – Gestion d'une base robotique « Rogue Blue »…… **107**
#26 – Gestion d'une base robotique type « Araignée »…… **111**

NOTE D'APPLICATION # 1.
BARGRAPH « LCD » de précision

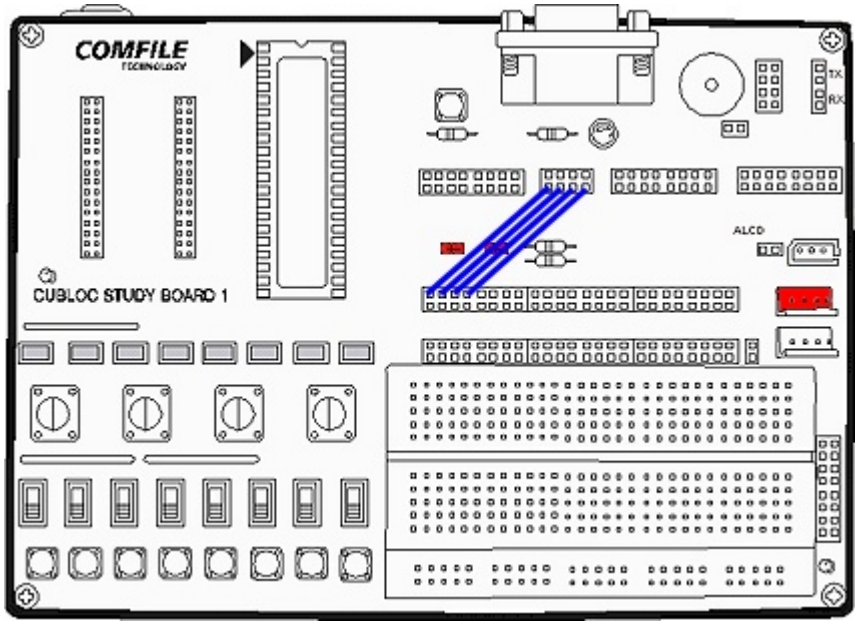
Cette note d'application va vous permettre de réaliser un bargraph de précision à usage universel sur un afficheur « LCD » alphanumérique. Ce bargraph pourra être utilisé pour représenter la variation d'une valeur numérique par le biais d'une barre de progression horizontale de longueur variable (plus la valeur numérique est importante, plus la barre est longue – et inversement).

Notions abordées :
- Gestion d'un écran « LCD » à commande I2C™ (avec redéfinition de caractères)
- Gestion d'entrées de conversion « Analogique / Numérique »
- Notion de « sous-routine »

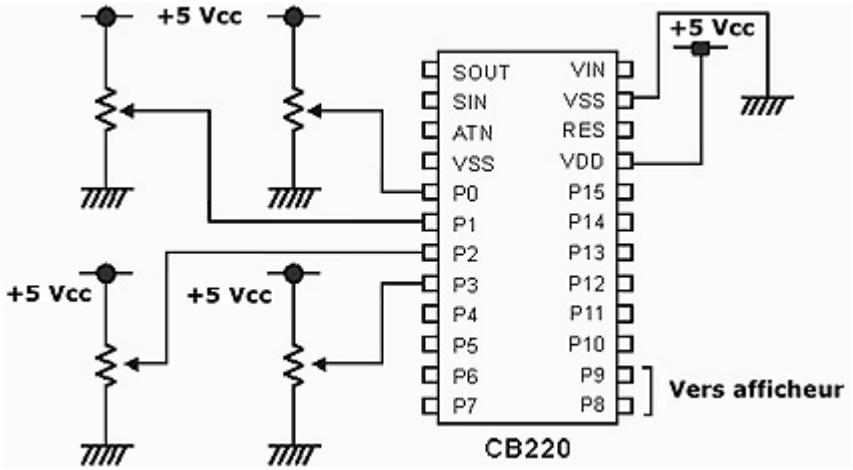
Matériel nécessaire :
- Une platine « CUBLOC Study Board » (facultative)
- 1 afficheur alphanumérique Comfile™ de type 2 x 16 ou 4 x 20 caractères à commande I2C™

Préparation matérielle :

Afin de faciliter la description de cette note d'application, nous utiliserons une platine « CUBLOC Study Board » associée à un module CUBLOC™ CB220. Commencez par relier l'afficheur LCD sur le port CuNET (marquage rouge) - Pensez à mettre les straps JP1 et JP2 afin de dédier les ports P8 et P9 au pilotage de l'afficheur (marquages rouges). Reliez ensuite les ports P0, P1, P2, P3 aux 4 curseurs de potentiomètres comme indiqué ci-dessous.

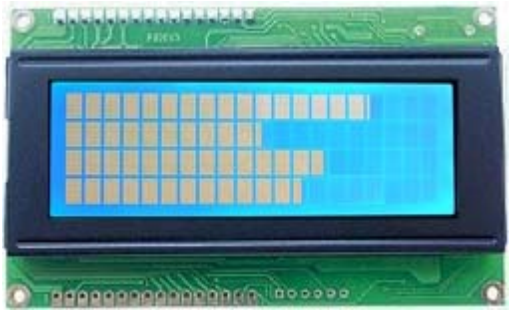


Le schéma ci-dessous permettra aux personnes ne disposant pas de platine « CUBLOC Study Board » de réaliser tout de même l'application.

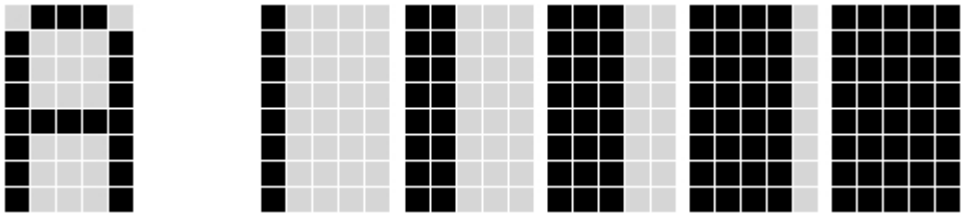


Principe général :

L'application consistera à récupérer les valeurs de tension présentes sur les curseurs des 4 potentiomètres de la platine afin d'en afficher une représentation indépendante sur 4 bargraphs horizontaux. La solution la plus simple pour réaliser les bargraph aurait été de faire afficher un caractère « plein » pour chaque unité de progression. Le problème majeur aurait alors été que nous aurions obtenu une résolution d'affichage très faible (20 pavés pour un afficheur de type 4 lignes de 20 caractères). C'est pourquoi la solution qui a été retenu est d'utiliser les possibilités de redéfinition de caractères des afficheurs LCD afin d'augmenter la précision d'affichage.



Chaque caractère de l'écran LCD alphanumérique est en fait constitué d'une matrice de 5 points horizontaux par 8 points verticaux. Parmi l'ensemble des caractères préexistants, il est possible de redéfinir soit même jusqu'à 8 caractères (en choisissant quels seront les points de la matrice à afficher). Dès lors il sera possible d'afficher de 1 à 5 lignes verticales sur une même matrice de l'afficheur.



A gauche le caractère « A » préexistant, à droite la représentation de 5 caractères redéfinis afin de pouvoir afficher une progression de 5 lignes verticales sur une même matrice. Cette méthode permettra ainsi d'obtenir des bargraphs avec une résolution d'affichage de 100 segments si vous utilisez un afficheur 4 lignes de 20 caractères (100 = 20 caractères x 5 lignes horizontales par caractères) ou une une résolution d'affichage de 80 segments si vous utilisez un afficheur 4 lignes de 16 caractères (80 = 16 caractères x 5 lignes horizontales par caractères).

Description détaillée du programme :

Le programme que vous retrouverez sur notre site Internet : www.lextronic.fr ou sur notre CD-ROM est disponible sous le nom « bargraph1 » (ce dernier permet la mise en œuvre d'un afficheur de type CLCD204-BLB de 4 x 20 caractères) – La version « bargraph2 » est conçue pour les utilisateurs d'afficheur de type CLCD162-BL de 2 x 16 caractères (dans ce cas, seul les 2 premiers potentiomètres génèrent l'affichage de bargraphs).

Le programme débute avec la déclaration du type de module CUBLOC™ utilisé (ici un CB220), puis avec l'initialisation de l'afficheur à commande I2C™ et son effacement (à ce titre, vérifiez que les commutateurs dils de l'afficheur soient tous sur « OFF » – sauf le dil 3 qui doit être sur « ON »). Les instructions **Delay** permettent de s'assurer que l'afficheur ai bien fini de s'effacer. Vous trouverez ensuite la phase de redéfinition de caractères qui permettra de « dessiner » et de préparer les lignes verticales (1 ligne, 2 lignes… 5 lignes) aux emplacements mémoire 8 à 12 de l'afficheur LCD (consultez la doc des afficheurs pour d'avantages d'informations sur la redéfinition de caractères). A noter que cette suite de redéfinition devra être ajoutée au début de votre programme si vous désirer exploiter le bargraph au sein d'une de vos applications. Le programme continue ensuite avec la déclaration des ports P0, P1, P2 et P3 en tant que port d'entrée (cette déclaration préalable est nécessaire puisque les ports vont être utilisés en tant qu'entrées de conversion analogique/numérique).

Le programme principal s'exécute au sein d'une boucle « sans fin » de type **do … loop** dans laquelle s'effectue la récupération successive des 4 valeurs analogiques présentent sur les curseurs des potentiomètres (sous la forme d'une variable **pot** de type Interger comprise entre 0 et 1023) afin d'en afficher une représentation sur les bargraphs. La génération des bargraphs associées à ces valeurs sera réalisée par une sous-routine que nous allons spécialement créer pour l'occasion. Les sous-routines sont des « morceaux » de programme qui pourront s'apparenter au final à une nouvelle instruction (ici **baragraph**). En faisant référence à cette instruction dans votre programme principal, vous exécuterez automatiquement la sous-routine en ayant la possibilité d'utiliser une variable de passage qui vous permettra d'indiquer sur quelle ligne (0 à 3) de l'afficheur vous voulez afficher le bargraph. Cette façon de programmer vous permettra de pouvoir vous créer une bibliothèque de « sous-routines » qu'il vous sera très facilement possible de réutiliser dans d'autres applications. Les « sous-routines » doivent toujours être déclarées après le programme principal (il vous faudra donc placer une instruction **end** après la dernière ligne de votre programme principal pour que le « Cubloc Studio » puisse savoir que la suite du programme correspond à une sous-routine).

Le bargraph à afficher est constitué de 3 parties calculées et générées au fur et à mesure :

- La première partie concernera le nombre de caractères plein à afficher (correspondant à un pavé de 5 segments horizontaux consécutifs sur une même matrice).

- La seconde partie concernera l'affichage d'une matrice sur laquelle il y a moins de 5 segments horizontaux.

- La troisième partie concernera l'affichage « d'espaces » destinés à effacer le reste de la ligne du bargraph (afin de rafraîchir ce dernier lorsque sa longueur diminue).

Les 3 parties sont « assemblées » les unes aux autres au sein d'une variable appelée barre (laquelle est de type « chaîne de caractère ») afin de pouvoir être affichée d'un coup.

La « sous routine » de génération des bargraphs commence par la déclaration des différentes variables qui seront utilisées. On mémorise ensuite dans les variables **valmax** et **i** le nombre de « pavés » plein à afficher (les pavés qui seront en fait composés de 5 segments horizontaux).

Pour ce faire, on divise la valeur analogique lue **pot** par le pas des segments 10,23 (10,3 = 1023 (valeur max. à afficher) / 100 segments)

Une fois ce nombre connu, on génère le début du bargraph en remplissant la variable **barre** à l'aide de **i** fois le caractère &HFF (qui correspond à un pavé plein). On calcule ensuite le nombre **J** de caractères « vides » qui devront être affichés à droite de la barre et on génère une variable **vide** (de type chaîne) avec autant de caractères «d'espace ».

On sélectionne enfin quel sera le caractère **segment** redéfini qui devra être affiché pour représenté une matrice où il y a moins de 5 segments horizontaux. On « additionne » ensuite les chaînes **barre** + **segment** + pour ensuite afficher l'ensemble en une seule fois sur la **ligne** désirée lors de l'appel à la sous routine.

Notes :
Si vous désirez utiliser la sous routine au sein d'une de vos applications, pensez à également ajouter les lignes de redéfinition de caractère au début de votre programme. Lorsque vous utilisez un afficheur 2 x 16 caractères certains paramètres de la « sous routine » changent afin d'adapter la résolution à 80 segments et non plus 100 comme avec un afficheur 4 x 20 caractères (voir le programme « bargraph2 »).

Il vous sera également possible d'utiliser des afficheurs de type « ALCD » (dans ce cas, vous devrez modifier les instructions de redéfinition de caractères et d'initialisation de l'afficheur afin de pouvoir exploiter ces derniers avec le port série). Le programme « bargraph3 » permet de remplacer la variation manuelle des 4 potentiomètres par une génération pseudo-aléatoire des valeurs associées à chaque bargraph de telle sorte que vous obteniez un effet très attractif (similaire aux vu-mètres d'un equalizeur).

Evolutions possibles :
Selon le même principe de redéfinition de caractères, il est possible d'envisager de modifier le programme afin de pouvoir bénéficier non plus de 4 bargraphs horizontaux, mais de 20 bargraphs verticaux composés chacun de 32 segments… A vous de « jouer » !

```
'###################################
'#         Gestion de BARGRAPHS          #
'#       sur un afficheur LCD 4 x 20         #
'#      @Lextronic 2006 - 19/01/2006       #
'###################################

Const Device = CB220                        ' Initialisation de l'afficheur LCD
Set Display 2,0,1,50
Dim pot As Integer
Delay 100
Cls
Delay 200
Csroff
Delay 100

Print &H1B,&H44,8,0,0,0,0,0,0,0             ' Redéfinition des caractères du LCD
Print &H1B,&H44,9,16,16,16,16,16,16,16,16
Print &H1B,&H44,10,24,24,24,24,24,24,24,24
Print &H1B,&H44,11,28,28,28,28,28,28,28,28
Print &H1B,&H44,12,30,30,30,30,30,30,30,30

Input 0                                     ' Configure ports conversion analogique/numérique en entrée
Input 1
Input 2
Input 3

Do
  pot = Adin(0)                     ' Lecture de la tension du potentiomètre 1
  bargraph 0                        ' Affichage de la valeur sous forme de bargraph sur la 1ère ligne
  pot = Adin(1)                     ' Lecture de la tension du potentiomètre 2
  bargraph 1                        ' Affichage de la valeur sous forme de bargraph sur la 2ème ligne
  pot = Adin(2)                     ' Lecture de la tension du potentiomètre 3
  bargraph 2                        ' Affichage de la valeur sous forme de bargraph sur la 3ème ligne
  pot = Adin(3)                     ' Lecture de la tension du potentiomètre 4
  bargraph 3                        ' Affichage de la valeur sous forme de bargraph sur la 4ème ligne

Loop
End


'###########################
'# Sous Routine BARGRAPH #
'###########################

Sub bargraph(ligne As Byte)
  Dim i As Byte
  Dim J As Byte
  Dim valmax As Single
  Dim barre As String*20
  Dim vide As String*20
  Dim segment As String*1
  valmax=(pot/10.23)/5
  i=valmax
  barre=String(&HFF,i)
  j=20-i
  vide=Spc(j)
  i=((valmax-i)/0.2)+8
  segment=Chr(i)
  barre=barre+segment+vide
  Locate 0,ligne
  Print barre
End Sub
```

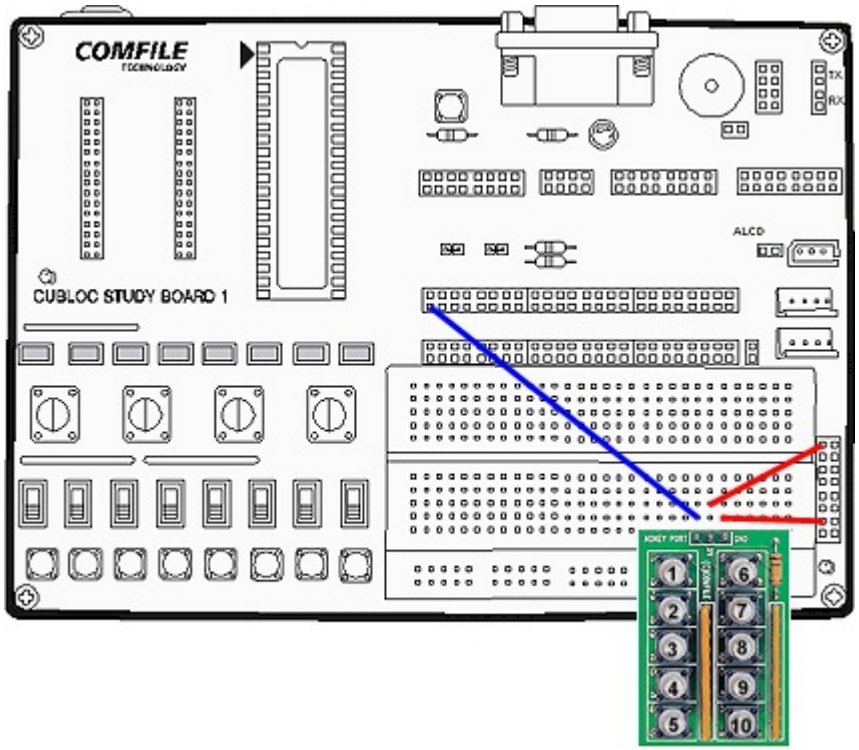
NOTE D'APPLICATION # 2.
Lecture de 10 boutons-poussoirs via une entrée

Cette note d'application va vous permettre de lire l'état de 10 boutons-poussoirs au moyen d'une petite platine appelée ADKEY, laquelle sera reliée sur une entrée de conversion « analogique / numérique » du CUBLOC™. Cette platine est en fait basée sur un principe de résistances « ponts diviseurs » que des boutons-poussoirs viennent shunter. En absence de sollicitation la tension en sortie de la platine est de +5 V. La tension prend des valeurs différentes en fonction de la touche qui est sollicitée. Il sera ainsi possible (suivant la valeur de la tension lue) de savoir quelle touche a été sollicitée. Dès lors vous pourrez connaître l'état de 80 boutons-poussoirs à l'aide de seulement 8 entrées de conversion « analogique/numérique » du CUBLOC™ !

Notions abordées :
- Gestion d'entrées de conversion « Analogique / Numérique »
- Création d'une « fonction »

Matériel nécessaire :
- Une platine « CUBLOC Study Board » (facultative) + 1 platine « ADKEY »

Préparation matérielle :
Le raccordement de la platine ADKEY se fera comme ci-dessous.



Le montage consistera simplement à alimenter la platine « ADKEY » et à relier sa sortie sur le port de conversion analogique/numérique P0 du CUBLOC™ CB220. La platine devra être câblée au plus près de l'entrée du module CUBLOC™.

Saisissez ensuite ce petit programme (ce dernier est disponible sur notre site Internet : www.lextronic.fr ou sur notre CD-ROM sous le nom « adkey1 »).

```
'#################################
'#  Gestion d'une platine ADKEY  #
'# @Lextronic 2006 - 28/01/2006  #
'#################################

Const Device = CB220
Dim touche As Integer

Input 0                                 ' Configure le port de conversion analogique/numérique en entrée

Do
  touche = Adin(0)                      ' Conversion de la tension disponible en sortie de la platine ADKEY
  If touche < 500 Then
      Debug Dec touche,Cr
  end if
Loop
```

Une fois exécuté, vous pourrez constater que ce programme vous permettra d'afficher la valeur de la conversion « analogique / numérique » relative à chaque touche sollicitée dans la fenêtre DEBUG du PC. Sur le schéma ci-dessus, nous avons numéroté les boutons de 1 à 10. Ainsi, la sollicitation du BP N° 1 retournera une valeur de l'ordre de 337, le BP N°2 une valeur de l'ordre de 379, BP N°3 -> 416, BP N° 4 -> 449, BP N° 5 -> 479, BP N° 6 -> 1, BP N° 7 -> 92, BP N° 8 -> 168, BP N° 9 -> 233 et BP N° 10 -> 288.

A noter que ces valeurs peuvent être quelque peu différentes (ceci est du aux tolérances des résistances utilisées sur les platines « ADKEY »). En absence de sollicitation, la valeur retournée est de l'ordre de 1023. Le test **if touche < 500** permet d'afficher les valeurs uniquement lorsqu'une touche est sollicitée (la valeur 500 a été choisie être sur que l'on se situe bien en dessous de 1023).

L'utilisation de ce petit programme amène plusieurs commentaires :

- En premier lieu, on pourra constater qu'il n'est pas possible de connaître l'état de plusieurs boutons-poussoirs sollicités en même temps.

- En second lieu vous pourrez remarquer que quelque fois des valeurs légèrement différentes peuvent s'afficher au moment où vous appuyez sur les poussoirs (ou au moment ou vous les relâchez). Ceci est du au fait que la tension générée par les réseaux de résistances via la sollicitation des boutons-poussoirs ne soit pas encore tout à fait stable au moment ou le CUBLOC™ effectue sa conversion « analogique/numérique ». L'effet rebond des poussoirs peut également interférer lors de la conversion « analogique/numérique ». Le second programme présenté ci-après va remédier à ces problèmes (ce dernier est disponible sur notre site Internet : www.lextronic.fr ou sur notre CD-ROM sous le nom « adkey2 »).

```
'#################################
'#      Gestion d'une platine ADKEY      #
'#          avec utilisation au sein          #
'#                d'une fonction                #
'#    @Lextronic 2006 - 28/01/2006      #
'#################################

Const Device = CB220
Dim bp As Integer

Do
   bp = adkey(bp)
   If bp <> 0 Then
     Debug Dec bp,Cr
   End If
Loop
End

'#########################
'#         Fonction ADKEY         #
'#########################

Function adkey (touche As Integer) As Integer
  Input 0                                  ' Configure le port de conversion analogique/numérique en entrée
  touche = Adin(0)                         ' Conversion de la tension disponible en sortie de la platine ADKEY
  adkey=0                                  ' on part du principe qu'aucune touche n'est sollicitée
  If touche < 500 Then
    Delay 10                               ' Temporisation permettant de s'assurer que la tension est stable
    touche = Adin(0)                       ' Conversion de la tension disponible en sortie de la platine ADKEY
    If touche < 485 Then adkey = 5         ' Décodage et attribution du N° de touche sollicité
    If touche < 455 Then adkey = 4
    If touche < 425 Then adkey = 3
    If touche < 385 Then adkey = 2
    If touche < 345 Then adkey = 1
    If touche < 295 Then adkey = 10
    If touche < 240 Then adkey = 9
    If touche < 175 Then adkey = 8
    If touche < 100 Then adkey = 7
    If touche < 10 Then adkey = 6
  End If
End Function
```

Pour ce nouveau programme, nous allons utiliser une possibilité très intéressante offerte par les CUBLOC™, laquelle vous permet d'avoir recours à la création de « fonctions ». Une « fonction » est une sous-routine qui pourra s'apparenter au final à une nouvelle instruction. En faisant référence à cette instruction dans votre programme principal, vous exécuterez automatiquement la sous-routine en récupérant le résultat de cette dernière. Cette façon de programmer vous permettra de pouvoir vous créer une bibliothèque de « fonctions » qu'il vous sera très facilement possible de réutiliser dans d'autres applications. Les « fonctions » doivent toujours être déclarées après le programme principal (il vous faudra donc placer une instruction **end** après la dernière ligne de votre programme principal pour que le « Cubloc Studio » puisse savoir que la suite du programme correspond à une « fonction »). Pour cette application, nous allons créer la fonction **adkey** dans laquelle nous utiliserons une variable **touche** de type Integer (laquelle sera déclarée dans l'entête de la fonction). Au cours de cette fonction, on commencera par initialiser la valeur adkey à zéro, puis à effectuer une conversion « analogique/numérique » de la tension présente sur le port P0 (que l'on aura au préalable défini en port d'entrée). Dès lors si une touche est sollicitée (si la valeur lue est < 500) on décodera son numéro – sans quoi la fonction s'achèvera et la valeur retournée sera zéro).

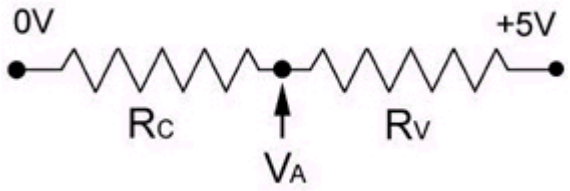
Le décodage consitera à tester sucessivement la valeur lue avec différents « paliers » présentant des valeurs intermédiaires par rapport aux valeurs relevées dans le premier programme (adkey1). En fonction de la tolérance des valeurs de résistances de la platine « adkey », vous pourrez être ammené à modifier légèrement ces valeurs intermédiaires.

Dans le programme principal qui s'effectue dans une boucle sans fin do …. Loop, on récupère simplement la valeur **adkey** de la fonction au sein d'une variable **bp**. Si la valeur n'est pas nulle (c'est à dire, si on a appuyé sur des touches), la valeur décodée de la touche s'affichera dans la fenêtre de debug du PC sous la forme d'un nombre compris entre 1 et 10).

Il vous sera dès lors possible d'utiliser des platines « adkey » associées au CUBLOC™ pour réaliser des systèmes de saisies sans monopoliser un grand nombre d'entrées/sorties.

NOTE D'APPLICATION # 3.
Gestion d'un capteur « FSR »

Cette note d'application va vous permettre d'interfacer un capteur de force « FSR » avec un module CUBLOC™. Les capteurs de force se caractérisent par le changement de leur valeur résistive sous l'action d'une force exercée sur leur surface. Plats et souples, pouvant être utilisés de -30°C à +70°C et dotés d'une durée de vie de l'ordre de 10 millions d'actons, ils se terminent par des broches à souder. Afin de pouvoir exploiter ces derniers, il conviendra de les associer à une résistance "talon" au sein d'un pont diviseur.



Notions abordées :
- Gestion d'un écran « LCD » à commande I2C™ (avec redéfinition de caractères)
- Gestion d'entrées de conversion « Analogique / Numérique »
- Notion de « sous-routine »

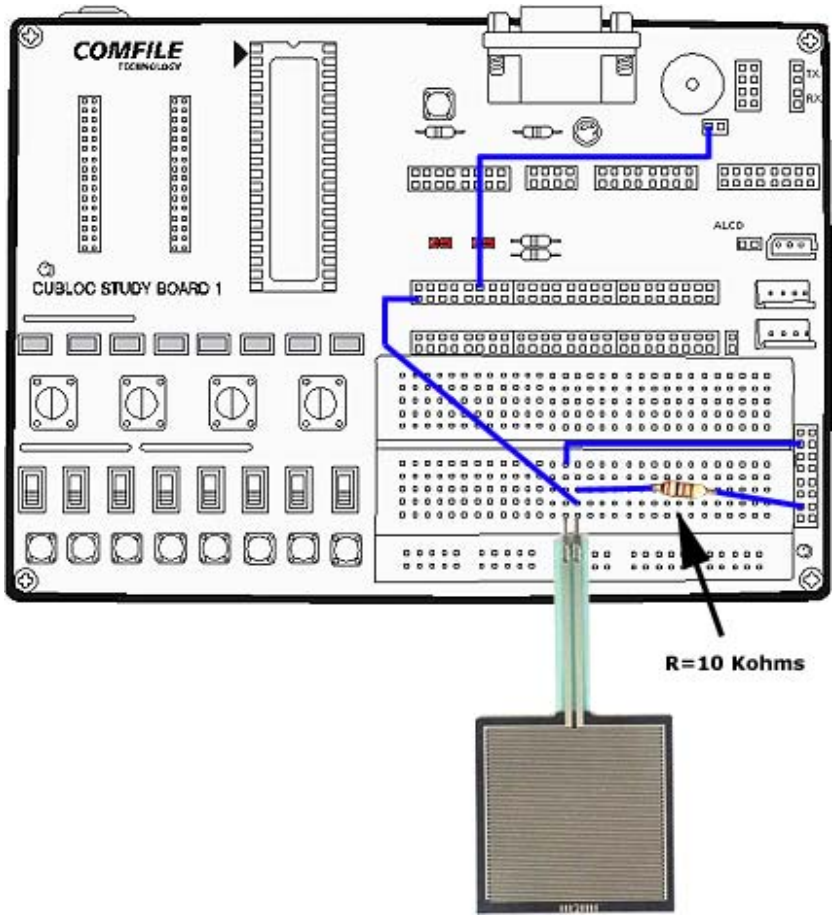
Matériel nécessaire :
- Une platine « CUBLOC Study Board » (facultative) + capteur « FSR »
- 1 afficheur alphanumérique Comfile™ de type 2 x 16 ou 4 x 20 caractères à commande I2C™

Ces capteurs pourront être utilisés pour une multitude d'applications. On prendra tout de même soins à ne pas plier et courber leur limande de façon excessive afin d'éviter de casser cette dernière et à souder les broches le plus rapidement possible. Le raccordement des capteurs FSR aux entrées des CUBLOC™ devra également être le plus court possible.

Nous disposons de 3 modèles type « bp » sous les références FSR1, FSR2, FSR3 et d'un modèle livré sous la forme d'une bande FSR4 (http://www.lextronic.fr/Capteurs/force.htm)

Préparation matérielle :

Suivant le schéma théorique vu ci avant, la mise en œuvre de ces capteurs nécessite l'emploi d'une résistance « talon » (RC) de 10 kohms que l'on ramènera à la masse tandis que le point milieu (VA) sera appliqué sur l'entrée de conversion « analogique/numérique » P0 du module CUBLOC™. On utilisera le port P5 en sortie afin de pouvoir solliciter le buzzer de la platine pour les besoins des applications décrites ci-après.




```
'###################################
'#        Gestion d'un Capteur FSR          #
'#     @Lextronic 2006 - 21/01/2006       #
'###################################

Const Device = CB220
Dim pot As Integer
Input 0                          ' Configure les ports de conversion analogique/numérique en entrée
Low 5                            ' Configure port P5 (PWM 0) en sortie et le force au niveau logique BAS
Do
   pot = Adin(0)                 ' Conversion de la tension reprise sur le FSR
   If pot > 10 Then
      Freqout 0,pot              ' Génère fréquence proportionnelle à la tension reprise sur le FSR
   Else
      Freqout 0,0
   End If
 Loop
```

Saisissez ensuite le petit programme présenté ci-dessus (ce dernier est disponible sur notre site Internet : www.lextronic.fr ou sur notre CD-ROM sous le nom « fsr1 »).

Ce programme permettra de générer un signal sonore sur le buzzer (dont la fréquence sera fonction de la pression que vous exercerez sur le capteur SFR). Le test **if pot > 10** permet d'éviter la génération de fréquence basse (crépitement) lorsque le capteur n'est pas sollicité). L'effet sonore est très original et vous pourrez facilement reproduire le sifflement d'un oiseau !

Vous disposez d'un second programme « fsr2 » sur notre site Internet : www.lextronic.fr ou sur notre CD-ROM, lequel reprend les mêmes possibilités que le programme précédent avec en plus l'affichage d'un bargraph sur un écran LCD spécialisé à commande I2C™. La progression du bargraph est proportionnelle à la pression exercée sur le capteur FSR. Consultez également la note d'application #1 pour plus d'informations sur le fonctionnement du barpgraph.

Vous disposez enfin d'un troisième programme « fsr3 » sur notre site Internet : www.lextronic.fr ou sur notre CD-ROM, lequel permet la réalisation d'une mini application mettant en œuvre le capteur « FSR » pour simuler un gradateur progressif » à commande manuelle d'une lampe avec fonction « ON/OFF » . Ce dernier permettra sur une pression faible sur le capteur « FSR » de générer la progression lente du bargraph (simulant ainsi par exemple l'allumage d'un éclairage avec une intensité de plus en plus importante). Si par contre vous réalisez une pression « forte » sur le capteur « FSR », le bargraph prend la valeur maximale (simulant l'éclairage maximal de l'éclairage). Une nouvelle pression faible sur le capteur « FSR » aura pour conséquence de diminuer progressivement le bargraph (en simulant ainsi l'extinction progressive de l'éclairage). A ce stade, si vous réalisez une pression « forte » sur le capteur « FSR », le bargraph prend la valeur minimale (simulant l'extinction totale de la lampe).

Pour résumer :
Pression « faible » sur le capteur -> Progression haute ou basse lente du bargraph.
Pression « forte » sur le capteur -> Activation « M/A » du bargraph (position max. / min).

La compréhension du fonctionnement de ce programme est relativement aisée. On utilisera une variable qui permettra de déterminer le sens de progression du bargraph (en monté ou en descente). On veillera à modifier ensuite le sens de progression du bargraph lorsque l'on aura détecté l'arrivée en buté « haute » ou « basse » de ce dernier.

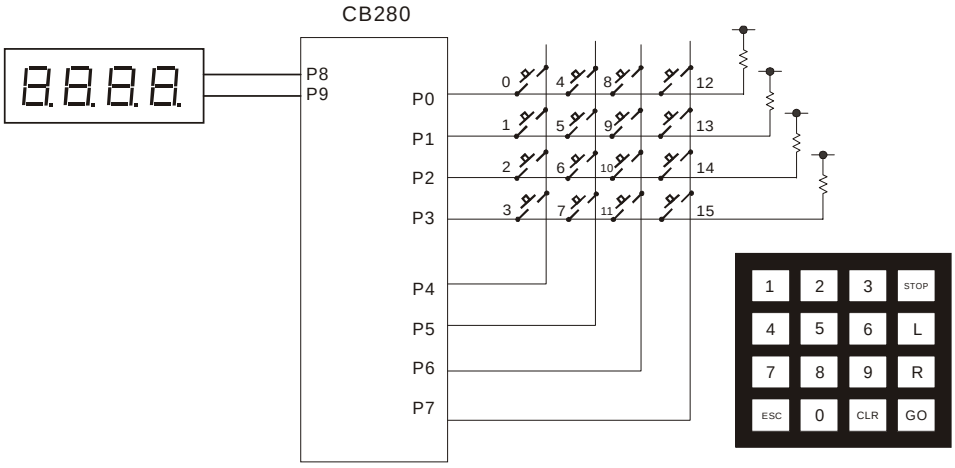
Lorsqu'on appuie fortement sur le capteur « FSR », le programme vérifiera que vous relâchiez la pression exercée sur e dernier afin d'éviter les activations « ON / OFF » en boucle.

```
'#################################
'#          Gestion d'un Capteur FSR          #
'#         et d'un afficheur LCD 4 x 20       #
'#          afin de réaliser un variateur     #
'#     progessif à commande manuelle     #
'#       @Lextronic 2006 - 22/01/2006       #
'#################################

Const Device = CB220
Set Display 2,0,1,50                                    ' Initialisation de l'afficheur LCD

Dim pot As Integer
Dim FSR As Integer
Dim sens As Long

Delay 100
Cls
Delay 200
Csroff
Delay 100

Print &H1B,&H44,8,0,0,0,0,0,0,0                         ' Redéfinition des caractères du bargraph
Print &H1B,&H44,9,16,16,16,16,16,16,16,16
Print &H1B,&H44,10,24,24,24,24,24,24,24,24
Print &H1B,&H44,11,28,28,28,28,28,28,28,28
 Print &H1B,&H44,12,30,30,30,30,30,30,30,30

Input 0                                 ' Configure le port de conversion analogique/numérique en entrée
Low 5                                   ' Configure port P5 (PWM 0) en sortie et le force au niveau logique BAS
pot = 0                                 ' Initialisation de la valeur de la graduation
sens=1

Do
    FSR = Adin(0)                       ' Conversion de la tension reprise sur le FSR1
    Do While FSR > 10                   ' Effectue les instructions comprises dans cette boucle si on appui sur le FSR
     If FSR < 800 Then   ' Si on appui doucement, augmente ou diminu le bargraph suivant sens de progesion initial
       pot = pot + sens
       If pot > 1023 Then pot = 1023    ' Le bargraph ne peut pas aller plus loin
       If pot < 0 Then pot = 0          ' Le bargraph ne peut pas aller plus bas
       bargraph 0                       ' Affichage de la valeur sous forme de bargraph sur la 1ère ligne
     Else
       If sens=1 Then                   ' On a appuye fort sur le SFR alors qu'on était en progession positive
         pot=1023                       ' le bargraph prend alors la valeur max.
         sens=-1                        ' Lors de la prochaine utilisation, le bargraph devra descendre
       Else                             ' On a appuyé fort sur le SFR alors qu'in était en progression négative
         pot=0                          ' le bargraph prend alors la valeur min.
         sens=1                         ' Lors de la prochaine utilisation, le bargraph devra monter
       End If
       bargraph 0                       ' Affichage de la valeur sous forme de bargraph sur la 1ère ligne
       Do While FSR > 10                ' Attend qu'on exerce plus de pression sur le SFR
         FSR = Adin(0)                  ' Lecture de la tension tension reprise sur le FSR
       Loop
     End If
     FSR=0                              ' Réinitialise la valeur lue via le FSR
     Loop
  Loop
  End

'#########################
'# Sous Routine BARGRAPH #
'#########################
```

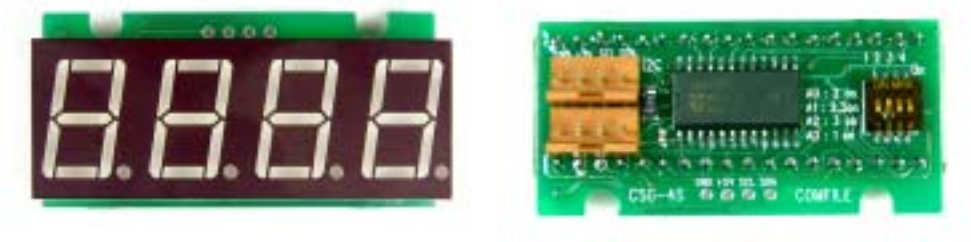
Cette sous-soutine est identique à celle de la note d'application #1

NOTE D'APPLICATION # 4.
Gestion d'un clavier matricé

Saisie sur un clavier matricé 16 touches (4 x 4) avec affichage sur module spécialisé 4 digits 7 segments à commande I2C™ (CSG)



Le module « CSG » est spécialement conçu pour être piloté via une liaison I2C™ au travers du port CUNET des modules CUBLOC™ afin que vous puissiez afficher des nombres et certaines lettres.



L'exemple ci-dessous vous permettra de faire afficher un compteur sur le module. Il est disponible sous le nom de fichier « csgprint ».

```
Const Device = CB280
Set I2c 9,8
Dim I As Byte
Do
  Csgdec 0,I
  I = I + 1
Loop
```

Le clavier matricé pourra être très facilement lu via la commande KEYPAD. Si vous regardez attentivement ce clavier, vous vous rendrez compte que l'ordre des touches ne correspond pas avec les N° de touches sollicitées. Afin de pouvoir avoir une lecture dans l'ordre, on aura recours à une table de correspondance avant d'afficher les valeurs sur le module « CSG ».

Ce programme est disponible sous le nom de fichier « keypadnum ».

```
Const Device = CB280
Set I2c 9,8
Dim I As Integer
Dim K As Integer

Const Byte KEYTABLE = (1,4,7,10,2,5,8,0,3,6,9,11,12,13,14,15)
Do
        I=Keypad(0)
        If I < 16 Then
                I = KEYTABLE(I)
                Csgdec 0,I
        End If
Loop
```

Le programme qui suit permet d'afficher la saisie dans la fenêtre Debug du PC . Lorsqu'une touche est sollicitée, elle est affichée sur le module « CSG » comme un nombre à 4 digits. Le nombre est stocké dans la variable K qui correspond à un code BCD. Nous utilisons ensuite la fonction BCD2BIN pour convertir à nouveau la valeur BCD en binaire. Ce programme est disponible sous le nom de fichier « num4in ».

```
Const Device = CB280
Set I2c 9,8
Dim I As Integer
Dim K As Integer
Dim M As Integer
K = 0
Const Byte KEYTABLE = (1,4,7,10,2,5,8,0,3,6,9,11,12,13,14,15)
Do
        I=Keypad(0)
        If I < 16 Then
                I = KEYTABLE(I)
                If I < 10 Then
                        K = K << 4
                        K = K + I
                        Csghex 0,K
                End If
                '
                '       WAIT UNTIL KEY DEPRESS
                '
                Do While Keypad(0) < 255
                Loop
                M = Bcd2bin(K)
                Debug Dec M,CR
        End If
Loop
```

Lorsqu'aucune touche du clavier n'est sollicitée, le code retourné par la commande KEYPAD aura pour valeur : 255. En utilisant **l'instruction Do While keypad(0) < 255**, nous pourrons attendre que toutes les touches soit relâchées (ce qui se traduira par l'obtention du code 255).

Ceci permet au module CUBLOC™ d'arrêter de lire les entrées pendant qu'une touche est sollicitée. Sans quoi, le processeur pourra recevoir plusieurs lectures de touche du fait de sa rapidité d'exécution.

En utilisant _D(0) = M, vous pourrez transférer la valeur de la touche dans le relay D0 du langage LADDER. Ainsi il vous sera possible de récupérer ce programme afin que vous puissiez utiliser un clavier de saisie en LADDER.

Nota : en cas d'intégration du clavier au sein d'une application, ce dernier devra être raccordé au plus près du module CUBLOC™.

NOTE D'APPLICATION # 5.
Gestion d'un télémètre ultrason « MSU08 »

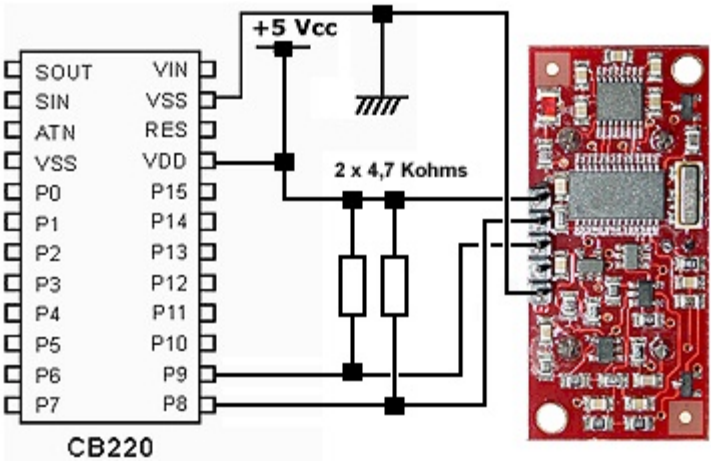
Cette note d'application va vous permettre de piloter un module télémètre ultrason « MSU08 ». Ce petit module, idéalement conçu pour les applications liées à la robotique ludique, est capable de déterminer la distance qui le sépare d'un obstacle se présentant devant lui (entre 3 cm et 6 m). Doté de 2 cellules ultrason, son principe de fonctionnement repose sur celui des "sonars". Il s'interface à l'aide de son bus I2C™ et se pilote à la manière d'une mémoire EEPROM type 24xx. Ce dernier peut vous retourner la valeur de la distance en "mm", en "inch" ou sous forme d'une durée (en µs) liée à l'echo de l'émetteur ultrason. A noter enfin qu'il vous sera possible d'adresser jusqu'à 16 modules différents par le bus I2C™. Il dispose également d'une LDR qui pourra également faire l'objet d'un mini détecteur de proximité. A noter que le « MSU08 » n'est destiné qu'à un usage ludique (pour la construction de petits robots mobiles capable d'éviter les obstacles).

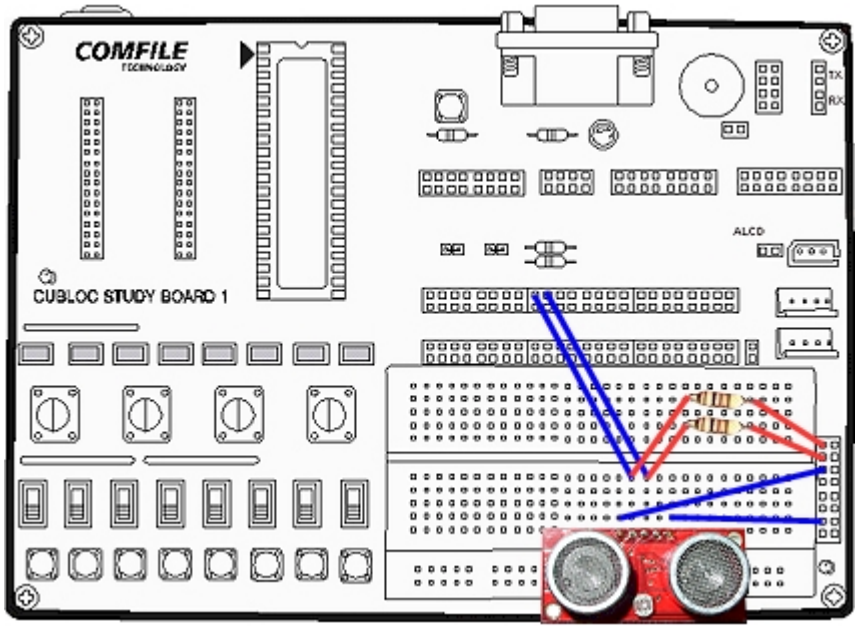
Notions abordées :
- Gestion I2C™

Matériel nécessaire :
- Une platine « CUBLOC Study Board » (facultative)
- 1 module « MSU08 »

Préparation matérielle :

Celle-ci repose sur le schéma théorique ci-dessous. Afin de faciliter la description de cette note d'application, nous utiliserons une platine « CUBLOC Study Board » associée à un module CUBLOC™ CB220 (voir schéma de raccordement ci-après).

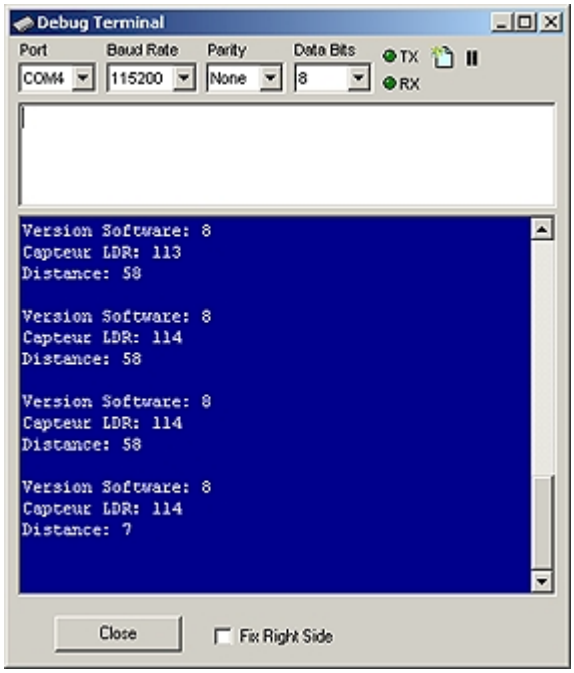




Les résistances ont pour valeur 4,7 Kohms.

Saisissez ensuite le petit programme présenté ci-dessus (ce dernier est disponible sur notre site Internet : www.lextronic.fr ou sur notre CD-ROM sous le nom « msu08 »).

Ce programme va permettre de venir lire les registres du module « MSU08 » après avoir initié une demande de mesure en cm. Dès lors, vous pourrez voir apparaître dans la fenêtre de DEBUG le N° du firmware du module « MSU08 », la valeur de la luminosité captée par la LDR du module ainsi que la distance en cm qui sépare le module d'un obstacle.



```
'###################################
'#   Gestion d'un module MSU08   #
'# @Lextronic 2006 - 03/02/2006  #
'###################################


Const Device = CB220
Dim errorcom As Byte
Dim version As Byte
Dim range As Integer
Dim ldr As Byte

Set I2c 8,9

Do
  I2cstart                      ' Condition Start I2C
  errorcom = I2cwrite (&HE0)    ' Adresse du module MSU08
  errorcom = I2cwrite (&H00)    ' Selectionne l'adresse du registre de commande
  errorcom = I2cwrite (&H51)    ' Active mesure en cm
  I2cstop                       ' Condition Stop I2C
  Delay 100                     ' Tempo
  I2cstart                      ' Condition Start I2C
  errorcom = I2cwrite (&HE0)    ' Adresse du module MSU08
  errorcom = I2cwrite (0)       ' Sélectionne l'adresse du premier registre à lire
  I2cstart                      ' Condition Start I2C
  errorcom = I2cwrite (&HE1)    ' Sélectionne condition de lecture I2C
  version = I2cread(0)          ' Récupere N° de révision du module MSU08
  errorcom = I2cwrite (&HE1)    ' Sélectionne condition de lecture I2C
  ldr = I2cread(0)              ' Récupere lecture valeur de la LDR
  errorcom = I2cwrite (&HE1)     ' Sélectionne condition de lecture I2C
  range.byte1=I2cread(0)        ' Récupère octet poids fort de la distance
  errorcom = I2cwrite (&HE1)     ' Sélectionne condition de lecture I2C
  range.byte0=I2cread(0)        ' Récupère octet poids faible de la distance
  I2cstop                       ' Condition Stop I2C

  Debug "Version Software: ",Dec version,Cr
  Debug "Capteur LDR: ",Dec ldr,Cr
  Debug "Distance: ",Dec range,Cr,Cr
  Delay 500
Loop
```

NOTE D'APPLICATION # 6.
Gestion d'une boussole électronique « CMP03 »

Cette note d'application va vous permettre de piloter un module Boussole électronique «CMP03 ». Ce petit module, idéalement conçu pour les applications liées à robotique ludique, est capable de détecter le "nord" grâce à l'emploi de 2 capteurs spécialisés montés en angle à 90° et par déduction de vous indiquer son orientation. Il s'interface à l'aide de son bus I2C™ et se pilote à la manière d'une mémoire EEPROM type 24xx. Ce dernier peut vous retourner une « image » de l'angle sous la forme d'un mot binaire compris entre 0 et 255 ou 0 et 3599 (correspondant à 0 et 359,9 °).

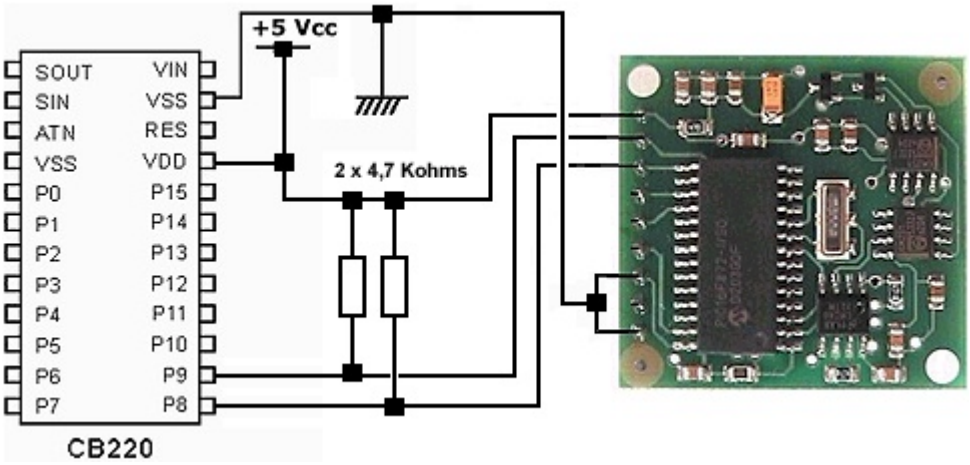
Afin d'augmenter la précision de la mesure, le module dispose également d'une procédure de calibrage et d'une intégration de l'influence des perturbations liées aux réseaux 50/60Hz.

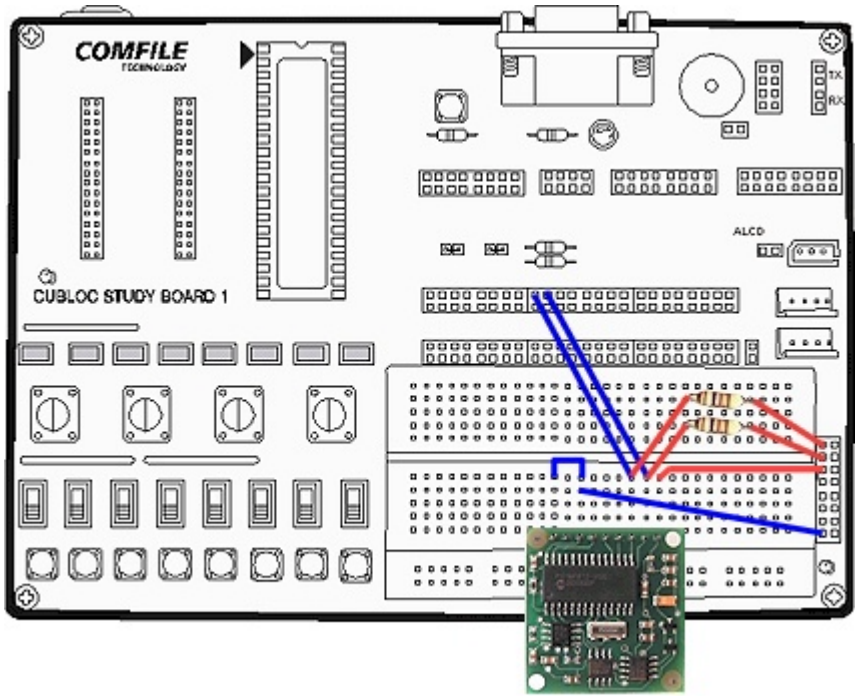
Notions abordées :
- Gestion I2C™

Matériel nécessaire :
- Une platine « CUBLOC Study Board » (facultative)
- 1 module « CMP03 »

Préparation matérielle :

Celle-ci repose sur le schéma théorique ci-dessous. Afin de faciliter la description de cette note d'application, nous utiliserons une platine « CUBLOC Study Board » associée à un module CUBLOC™ CB220 (voir schéma de raccordement ci-après).

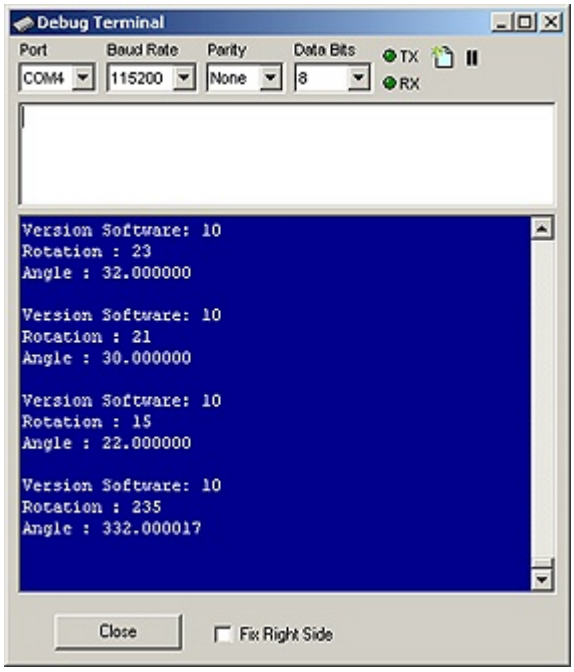




Les résistances ont pour valeur 4,7 Kohms.

Saisissez ensuite le petit programme présenté ci-dessus (ce dernier est disponible sur notre site Internet : www.lextronic.fr ou sur notre CD-ROM sous le nom « cmp03 »).

Ce programme va permettre de venir lire les registres du module « cmp03 ». Dès lors, vous pourrez voir apparaître dans la fenêtre de DEBUG le N° du firmware du module « cmp03 », l'image de l'angle mesuré (sous la forme d'un nombre compris entre 0 et 255) et la valeur directe de l'angle (sous la forme d'un nombre compris entre 0 et 359,9).



```
'###################################
'#        Gestion d'un module CMP03      #
'#      @Lextronic 2006 - 14/02/2006     #
'###################################


Const Device = CB220
Dim errorcom As Byte
Dim version As Byte
Dim compass1 As Byte
Dim compass2 As Integer
Dim angle As Single

Set I2c 8,9

Do
   I2cstart                                ' Condition Start I2C
   errorcom = I2cwrite (&HC0)              ' Adresse du module CMP03
   errorcom = I2cwrite (0)                 ' Sélectionne l'adresse du premier registre à lire
   I2cstart                                ' Condition Start I2C
   errorcom = I2cwrite (&HC1)              ' Sélectionne condition de lecture I2C
   version = I2cread(0)                    ' Récupere N° de révision du module CMP03
   errorcom = I2cwrite (&HC1)              ' Sélectionne condition de lecture I2C
   compass1 = I2cread(0)                   ' Récupere lecture valeur angle (0 - 255)
   errorcom = I2cwrite (&HC1)              ' Sélectionne condition de lecture I2C
   compass2.byte1=I2cread(0)               ' Récupère octet poids fort de la distance
   errorcom = I2cwrite (&HC1)              ' Sélectionne condition de lecture I2C
   compass2.byte0=I2cread(0)               ' Récupère octet poids faible de la distance
   I2cstop                                 ' Condition Stop I2C

   angle= compass2/10                      ' division par 10 pour affichage 0 - 359,9
   Debug "Version Software: ",Dec version,Cr
   Debug "Rotation : ",Dec compass1,Cr
   Debug "Angle : ",Float angle,Cr,Cr
   Delay 500
 Loop
```

NOTE D'APPLICATION # 7.
Gestion afficheur à dalle tactile « ezDISPLAY »

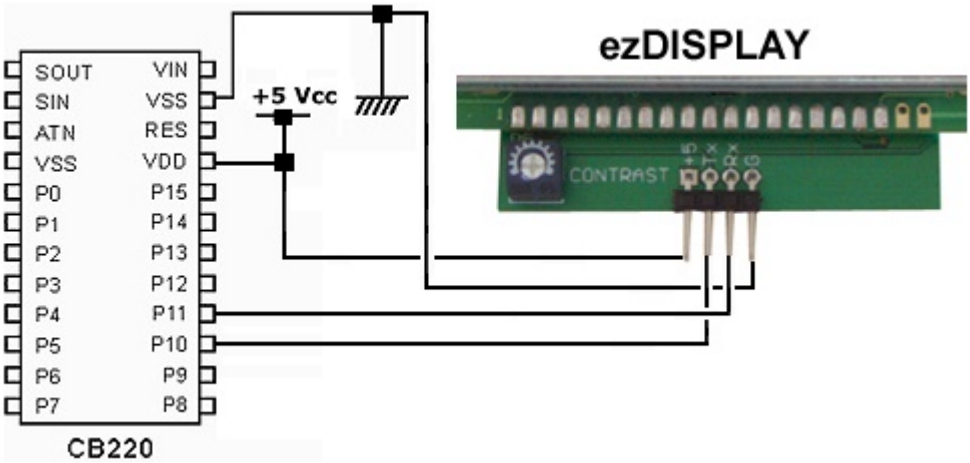
Cette note d'application va vous permettre de piloter un afficheur LCD graphique 180 x 80 pixels à dalle tactile « ezDSIPLAY ». Ce dernier vous permettra d'écrire du texte, de tracer des lignes droites horizontales ou verticales ou encore de mémoriser 4 images « BMP » à partir d'un PC pour les rappeler à l'écran à tout moment. Economique et simple d'utilisation il se pilote via une liaison série 9600 bds (TX / RX) au niveau logique + 5 V. A noter que cet afficheur n'est pas directement compatible avec les instructions de gestion des afficheurs LCD graphiques propres aux modules CUBLOC. Note : cet afficheur est désormais dispo en version rétro-éclairée (le modèle de cette note d'application est une ancienne version non rétro-éclairée).

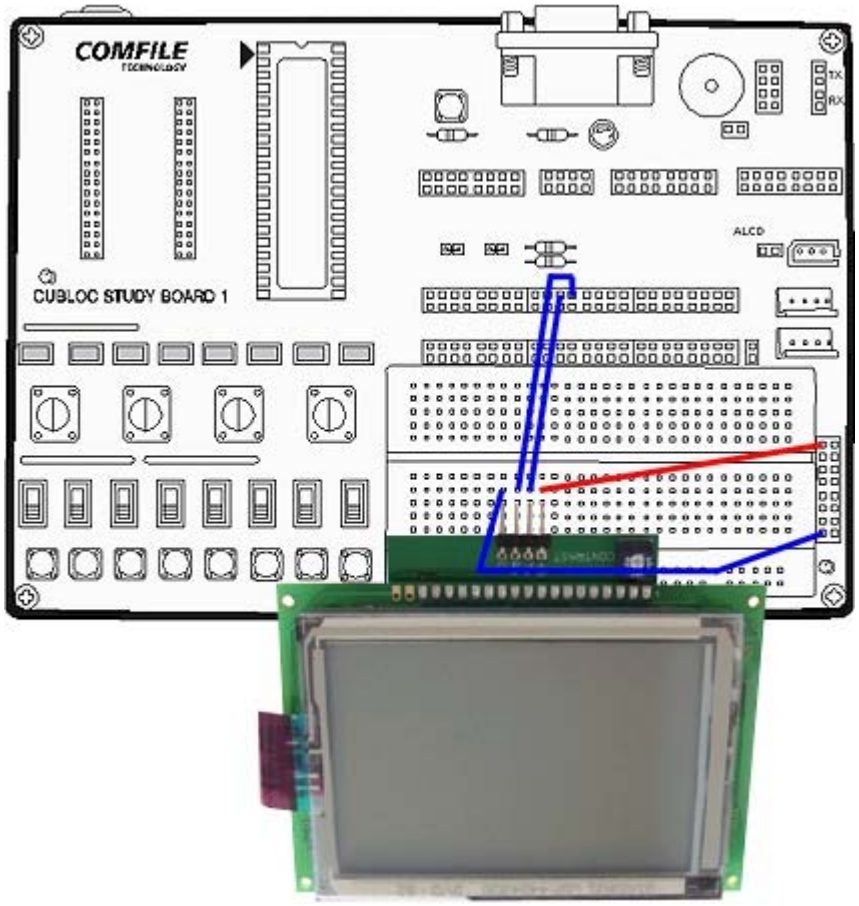
Notions abordées :
- Gestion d'une communication série

Matériel nécessaire :
- Une platine « CUBLOC Study Board » (facultative)
- 1 module « ezDISPLAY »

Préparation matérielle :

Celle-ci repose sur le schéma théorique ci-dessous.



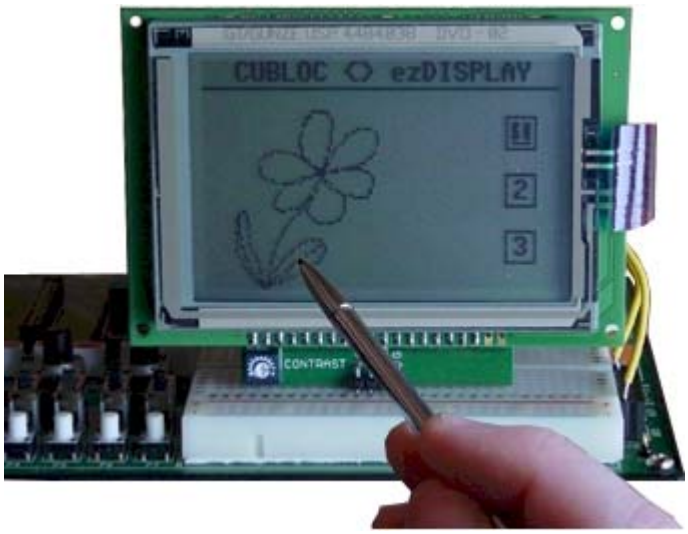


Si vous utilisez un CB280, il vous faudra modifier le câblage de la liaison série en reliant la broche TX de l'afficheur « ezDISPLAY » à la borne RX (CB280) de la platine « CUBLOC Study Board » (présente en haut à droite de celle-ci) et la borne RX de l'afficheur « ezDISPLAY » à la borne TX (CB280) de la platine « CUBLOC Study Board ».

A noter que lors de l'intégration finale au sein de votre application, l'afficheur devra être relié au plus près du module CUBLOC™ (il ne faudra pas le déporter avec une grande longueur de câble). Son usage se limite aux applications domestiques (il n'est pas conçu, ni prévu pour être utilisé au sein d'applications embarquées dans des véhicules – consultez sa notice pour prendre connaissance des limites d'utilisation en température et autres spécifications).

Saisissez ensuite le petit programme présenté ci-après (ce dernier est disponible sur notre site Internet : www.lextronic.fr ou sur notre CD-ROM sous le nom « ezdisplay »). Il vous permettra d'apprendre à écrire du texte sur l'écran, à tracer des lignes, à faire apparaître et disparaître des pixels, à effacer tout ou partie de l'écran et à récupérer les informations de la dalle tactile.

A ce titre il vous faudra vous munir d'un petit « stylet » à tête caoutchouc pour solliciter la dalle tactile (tels que ceux que l'on trouve dans les PDA par exemple). Si vous n'en disposez pas, vous pourrez au besoin vous munir d'un capuchon de stylo par exemple. Dans tous les cas il vous faudra manipuler ce dernier avec attention et ne pas exercer une pression trop forte sur l'écran.



Lorsque vous exécutez le programme, l'afficheur inscrit un petit texte de présentation en haut de l'écran avec 3 boutons de sélection sur le côté droit. Le bouton « 1 » est grisé (affichage inversé). Ceci signifie que vous êtes en mode « dessin » et qu'il vous est possible en déplaçant le stylet sur l'écran (de façon lente) de laisser une « trace » à l'écran par le biais de pixels qui s'affichent. En sollicitant le bouton « 2 » avec le stylet, le bouton « 1 » se désactive (le chiffre 1 n'est plus inscrit en inversé) et c'est le bouton « 2 » qui est sélectionné (ce dernier s'affiche en inversé). A ce stade, vous être en mode « gomme » et lorsque vous déplacez le stylet sur des pixels, ces derniers s'effacent. En sollicitant le bouton « 3 », vous effacez tous vos tracés et l'afficheur repasse en mode « 1 » (dessin).

Interprétation du programme :
Après la définition des variables, le programme démarre par l'initialisation du port série du CUBLOC™ (9600 bds/8bits/sans parité/1 stop). La variable « mode » servira à mémoriser le type d'affichage que l'on effectuera avec le stylet sur les pixels de l'afficheur (si mode = 1 -> on affiche les pixels, si mode = 0 -> on efface les pixels). 2 fonctions ont été crées pour cette note d'application afin de vous permettre d'utiliser encore plu facilement l'afficheur.

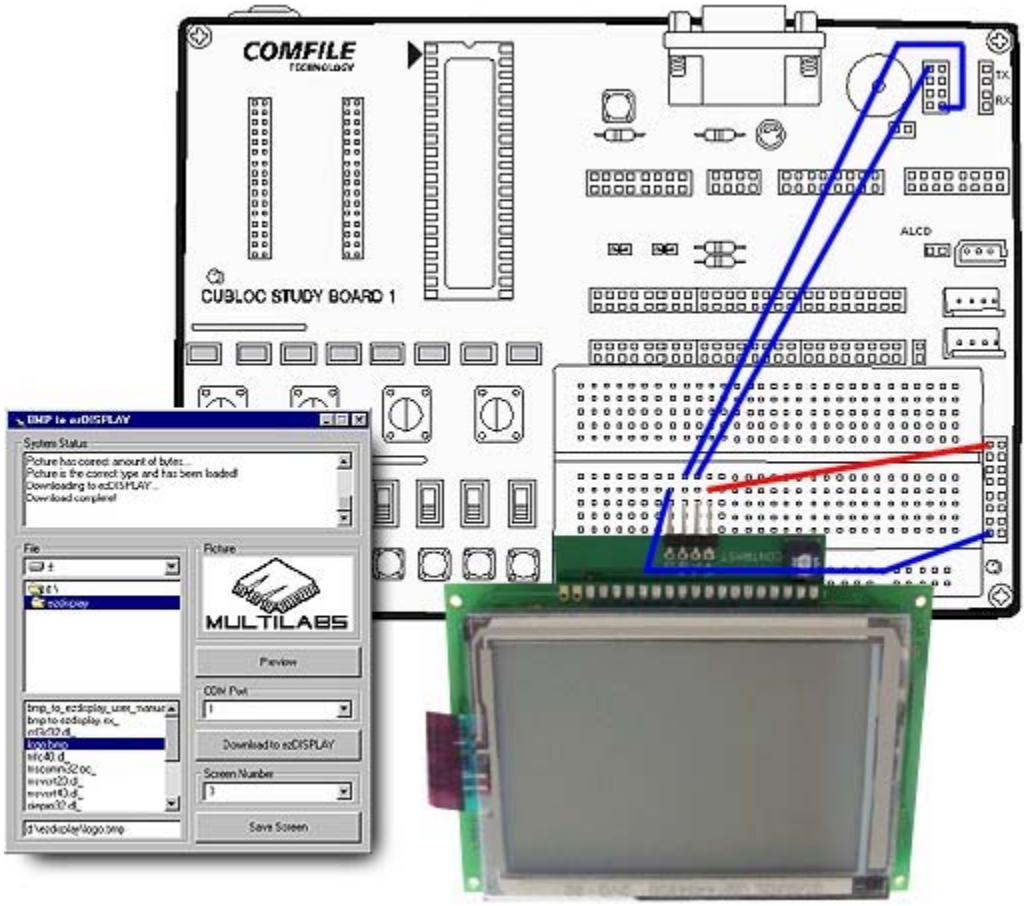
La première fonction « ezprint » permet d'afficher du texte à l'écran. Celle-ci nécessite que vous déclarez le texte (dans la variable « texte »), puis que vous appeliez la fonction en spécifiant les coordonnées (x , y) où le texte doit s'afficher (x =1 à 180 et y = 1 à 80). Attention toutefois à sélectionner un texte et des coordonnées « valides » afin que votre texte ne sorte pas de la fenêtre de l'afficheur (au besoin il est très facilement possible d'inclure un test conditionnel sur les valeurs x et y dans la fonction « ezprint » afin que rien ne soit envoyé à l'afficheur en cas de tentative d'affichage en dehors de sa fenêtre). La fonction « ezprint » est très simple à comprendre. Celle-ci consiste à récupérer un à un les caractères de la variable « texte » afin de les afficher à l'écran en prenant soin d'incrémenter la coordonnée X de 7 unités pour le prochain caractère. En augmentant ou en diminuant cette valeur, vous pourrez inscrire un texte plus ou moins « serré » à l'écran.

La seconde fonction « ezline » permet de tracer une ligne horizontale ou verticale en spécifiant les coordonnées x,y de départ et d'arrivée de celle-ci. Ces 2 fonctions sont mises à contribution pour afficher l'écran de présentation et les boutons sur l'écran. L'affichage des chiffres présent à l'intérieur des boutons est quant à lui effectué en envoyant directement des ordres à l'afficheur (sans utiliser la fonction « ezprint ») – Ceci est du au fait que certains chiffres nécessitent d'être affiché en « vidéo inverse ». Le programme principal consiste à récupérer les coordonnées (x, y) de la dalle tactile et en fonction de leur position, d'afficher (ou d'éteindre) le pixel présent sous le stylet, ou si le stylet a sollicité un des 3 boutons de l'afficheur, de changer de mode (allumage /extinction de pixel – ou effacement d'une partie de l'écran). On remarquera que l'allumage/ l'extinction des pixels ne peut se faire que sur une partie de l'écran (grâce à des tests conditionnels) afin d'éviter que vous n'effacer ou que vous ne dessiniez sur le texte de présentation ou les boutons-poussoirs.

```
'###################################
'#  Gestion d'un afficheur LCD     #
'#  à dalles tactiles ezDISPLAY    #
'# @Lextronic 2006 - 15/04/2006    #
'###################################

Const Device = CB220
Dim mode As Byte                          ‘ Variable de sélection de mode
Dim x As Byte
Dim y As Byte
Dim texte As String*23
Dim position As Integer
Opencom 1,9600,3,30,20
Delay 100
mode = 1
Putstr 1,"E",1                            ' Effacement de tout l'écran
Delay 500
texte="CUBLOC <> ezDISPLAY"               ' Affiche le titre
ezprint 15,1
ezline 1,10,160,10                        ' Trace ligne horizontale

‘ ********** Affiche le nom des BP (1 – 2 – 3 ) *************
Putstr 1,67,2,"1",140,22
Delay 15
Putstr 1,67,0,"2",140,42
Delay 15
Putstr 1,67,0,"3",140,62
Delay 15

‘ *************** Trace les contours des BP *************
ezline 137,20,137,31                      ' Trace contours des BP
ezline 137,40,137,51
ezline 137,60,137,71
ezline 150,20,150,31
ezline 150,40,150,51
ezline 150,60,150,71
ezline 137,20,150,20
ezline 137,40,150,40
ezline 137,60,150,60
ezline 137,31,150,31
ezline 137,51,150,51
ezline 137,71,150,71
Bclr 1,0                                  ' Efface buffer réception

‘ ******************** Boucle principale ******************
Do
  Putstr 1,"G"                            ‘ Récupère position du stylet
  position=Get(1,2)
  x=position.byte0
  y=position.byte1
  If x>0 And x<129 And y>15 Then          ' Test si on peut tracer au stylet
    Putstr 1,"P",mode,x,y
    Delay 15
    Bclr 1,0
  End If
```

```
If x>137 And x<150 And y>20 And y<51 Then    ' test si appuie sur le BP 1 ?
    mode = 1
    Putstr 1,67,2,"1",140,22                 ' Affiche 1, 2
    Delay 15
    Putstr 1,67,0,"2",140,42
    Delay 15
    Bclr 1,0
End If

If x>137 And x<150 And y>40 And y<51 Then    ' test si appui sur le BP 2 ?
    mode = 0
    Putstr 1,67,0,"1",140,22                 ' Affiche 1, 2
    Delay 15
    Putstr 1,67,2,"2",140,42
    Delay 15
    Bclr 1,0
End If

If x>137 And x<150 And y>60 And y<71 Then    ' test si appui sur le BP 3 ?
    Putstr 1,"E",0,1,16,128,80
    Delay 800
    mode = 1
    Putstr 1,67,2,"1",140,22                                    ' Affiche 1, 2
    Delay 15
    Putstr 1,67,0,"2",140,42
    Delay 15
    Bclr 1,0
End If

Delay 10
Loop
End

'##########################
'#     Sous Routine ezPRINT    #
'##########################
Sub ezprint(x1 As Byte,y1 As Byte)
   Dim z As Byte
   Dim caract As String*1
           For z = 1 To Len(texte)
      caract=Mid(texte,z,1)
      Putstr 1,67,0,caract,x1,y1
      x1=x1+7
      Delay 25
   Next
  End Sub


'##########################
'# Sous Routine ezLINE   #
'##########################
Sub ezline(x1 As Byte,y1 As Byte,x2 As Byte,y2 As Byte)
     Putstr 1,"L",1,x1,y1,x2,y2
     Delay 25
   End Sub
```

Astuce :
Il est possible de télécharger de 1 à 4 images « BMP » au sein de l'afficheur « ezDISPLAY » depuis un compatible PC via le port série à l'aide d'un utilitaire spécial. Ces images peuvent être sauvegardées dans la mémoire de l'afficheur afin de pouvoir être affichées à tout moment par la suite par votre module CUBLOC™ (consultez la notice de l'afficheur pour plus d'information). Pour réaliser le transfert des images depuis le PC vers l'afficheur il vous faut par contre réaliser un montage d'adaptation pour mettre à niveau les signaux +/- 12 V de la liaison RS232 du PC avec les signaux série 0/5V de l'afficheur. Si vous utilisez une platine « CUBLOC Study Board », il vous sera très facilement possible de mettre à profit le convertisseur intégré de la platine pour cette opération. Réalisez alors le montage ci-dessous. Reliez ensuite le câble série du PC non pas sur le connecteur sub-D 9 broches (DOWNLOAD) du haut de la platine, mais sur celui présent au dos dernière la platine (à côté de la prise d'alimentation DC 9 V) et exécutez tout simplement le programme de téléchargement de l'afficheur « ezDISPLAY ».



NOTE D'APPLICATION # 8.
Gestion module synthétiseur vocal « SP03 »

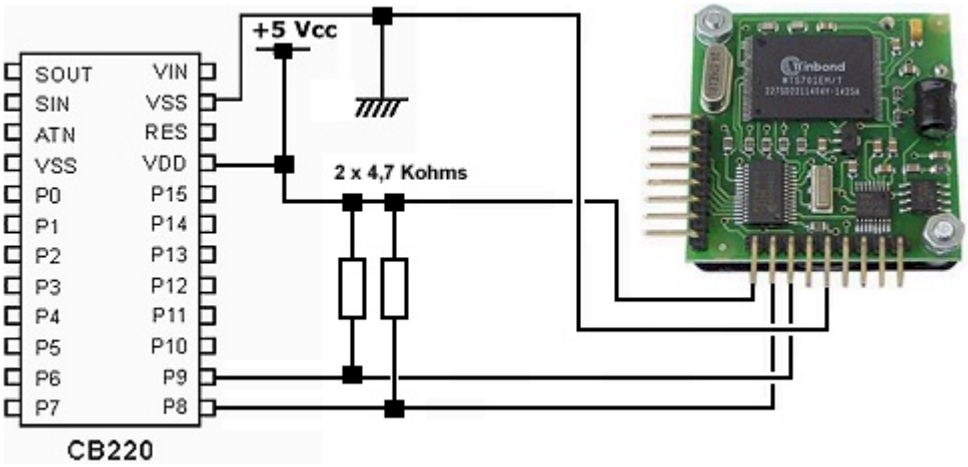
Cette note d'application va vous permettre de piloter un petit module spécialisé capable de prononcer des phrases (**en Anglais**) qui lui auront été transmises sous forme de «texte» via un bus I2C™. Conçu sur la base d'un processeur Winbond™ WTS701, le module intègre un amplificateur audio, un étage de régulation (3 V), un microcontrôleur PIC qui vous permettra de disposer d'un mode de pilotage très aisé et d'un haut-parleur de 40 mm.

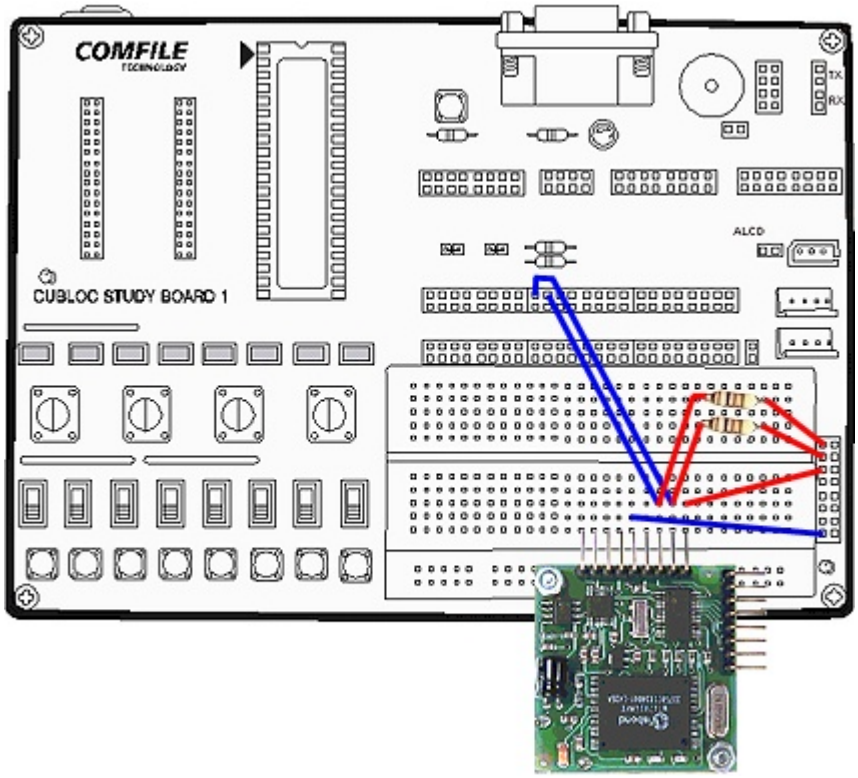
Notions abordées :
- Gestion d'une communication I2C™.

Matériel nécessaire :
- Une platine « CUBLOC Study Board » (facultative)
- 1 module « ezDISPLAY »

Préparation matérielle :

Celle-ci repose sur le schéma théorique ci-dessous. Les ports P8 (SDA) et P9 (SCL) sont dédiés à la communication I2C™ avec le module SP03.





Les résistances ont pour valeur 4,7 Kohms.

Saisissez ensuite le petit programme présenté ci-dessus (ce dernier est disponible sur notre site Internet : www.lextronic.fr ou sur notre CD-ROM sous le nom « SP03 »).

Pour les besoins de la note d'application, nous avons créé 2 fonctions. La première « play » vous permettra de prononcer des phrases directement en les écrivant dans la variable « texte ». Lors de l'appel de la fonction « play », les 3 paramètres envoyés correspondent respectivement au volume sonore (0 – fort à 7 – faible), au pitch (0 à 7) et à la vitesse (0 à 3) de restitution de la phrase. Vous pourrez ainsi essayer différentes combinaisons pour arriver au résultat le plus harmonieux. La fonction « play » va alors envoyer ces informations au module « SP03 » via le bus I2C™, puis ensuite envoyer les lettres de votre phrase une part une avant de « clore » le message et d'envoyer un nouvel ordre pour que le module prononce la phrase. Une boucle spéciale vient ensuite lire le registre principal du module « SP03 » afin d'attendre que ce dernier prenne la valeur 0 (ce qui signifiera que le message n'est plus prononcé et que le message peut recevoir un autre ordre).

La seconde fonction « replay » permet pour sa part de restituer de 1 à 30 messages que vous pourrez préalablement enregistrer dans le module via une liaison série annexe (voir notice du module vocal). Ici le module restituera le message N°1 pré-enregistré dans chaque module livré « Thank's for processing the SP03 ». Ceci permet de limiter la taille de votre programme si vous avez des messages « types » devant être répétés souvent. A noter que la procédure d'attente de fin de prononciation a également été recopiée dans la fonction « replay » (rien ne vous empêche de la gérer en tant que sous-routine à part entière en la retirant des fonctions et de l'appelant au retour des fonctions « play » et « replay ». A noter enfin qu'à la mise sous tension du module « SP03 », le message pré-enregistré N°1 sera systématiquement et automatiquement énoncé par le « SP03 » (n'interprétez donc pas celà comme un dysfonctionneùent de votre module CUBLOC).

```
'#################################
'#         Gestion d'un module SP03         #
'#     @Lextronic 2006 - 20/02/2006         #
'#################################

Const Device = CB220
Dim errorcom As Byte
Dim Texte As String * 60

Set I2c 8,9
Delay 100

Do
  texte="hello ! my CB220 module is very fantastic"
  play 0,5,1
  Delay 1000
  texte="This is a new interresting application"
  play 0,5,1
  Delay 1000
  replay 1                              ' restitution du message N° 1 pré-chargé en mémoire
  Delay 1000
 Loop
 End


'############################
'#         Sous Routine play        #
'############################

Sub play (volume As Byte, pitch As Byte, vitesse As Byte)
  Dim z As Byte
  Dim caract As String*1
  I2cstart                              ' Condition Start I2C
  errorcom = I2cwrite (&HC4)            ' Adresse du module SP03
  errorcom = I2cwrite (&H00)            ' Adresse registre de commande
  errorcom = I2cwrite (&H00)            ' Commande NOP -> Début message
  errorcom = I2cwrite (volume)          ' Volume sonore
  errorcom = I2cwrite (pitch)           ' Pitch
  errorcom = I2cwrite (Vitesse)         ' Vitesse
  For z = 1 To Len(texte)
    caract=Mid(texte,z,1)
    errorcom = I2cwrite (Asc(caract))
  Next
  errorcom = I2cwrite (&H00)            ' Fin message
  I2cstop                               ' Condition Stop I2C
  I2cstart                              ' Condition Start I2C
  errorcom = I2cwrite (&HC4)            ' Adresse du module SP03
  errorcom = I2cwrite (&H00)            ' Adresse registre de commande
  errorcom = I2cwrite (&H40)            ' Prononce la phrase
  I2cstop                               ' Condition Stop I2C

 '*************** Attend que le SP03 arrête de parler **********
 Do While z<>0
    I2cstart                            ' Condition Start I2C
    errorcom = I2cwrite (&HC4)          ' Adresse du module SP03
    errorcom = I2cwrite (&H00)          ' Adresse registre de commande
    I2cstart                            ' Condition Start I2C
    errorcom = I2cwrite (&HC5)          ' Sélectionne condition de lecture I2C
    z = I2cread(0)                      ' Récupere la valeur du registre de commande
    I2cstop                             ' Condition Stop I2C
 Loop
 End Sub
```

```
'#########################
'#       Sous Routine replay     #
'#########################

 Sub replay (nb As Byte)
    I2cstart                        ' Condition Start I2C
    errorcom = I2cwrite (&HC4)      ' Adresse du module SP03
    errorcom = I2cwrite (&H00)      ' Adresse registre de commande
    errorcom = I2cwrite (nb)        ' Restitution N° message pré-mémorisé
    I2cstop                         ' Condition Stop I2C

    '*************** Attend que le SP03 arrête de parler **********
    Do While nb<>0
       I2cstart                     ' Condition Start I2C
       errorcom = I2cwrite (&HC4)   ' Adresse du module SP03
       errorcom = I2cwrite (&H00)   ' Adresse registre de commande
       I2cstart                     ' Condition Start I2C
       errorcom = I2cwrite (&HC5)   ' Sélectionne condition de lecture I2C
       nb = I2cread(0)              ' Récupere la valeur du registre de commande
       I2cstop                      ' Condition Stop I2C
     Loop
  End Sub
```

NOTE D'APPLICATION # 9.
Gestion d'un capteur thermique « MTP81 »

Cette note d'application va vous permettre de piloter un capteur thermique pouvant s'apparenter à une mini-caméra thermique basse résolution. A l'opposé des cellules infrarouges passives, lesquelles sont uniquement capables de déceler des mouvements hu-mains (par détection de la différence de température entre le rayonnement infrarouge du corps humain et la température de la pièce), le capteur utilisé sur le module "MTP81" est à même d'effectuer des mesures directes de température (comme sur les thermomètres infrarouges sans contact).

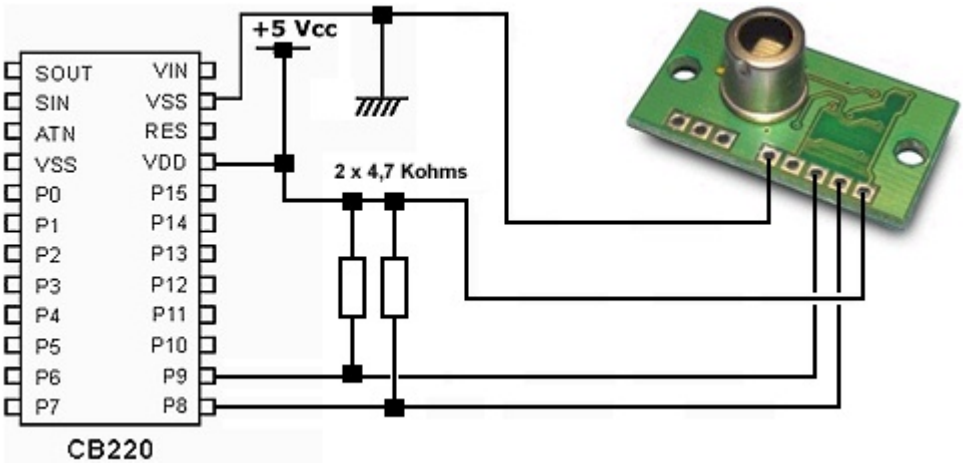
Le module est ainsi doté de 8 mini-zones sensibles consécutives capables de mesurer la température d'un point donné. Destiné à être alimenté sous +5V, le module "MTP81" se pilote au moyen d'un Bus I2C™. Une sortie spéciale permet de piloter automatiquement un servomoteur (type modélisme) afin que le module (devant être monté sur le palonnier du servo) puisse balayer une zone complète pour obtenir le spectre thermique d'une large surface – Cette possibilité n'est pas exploitée dans cette note d'application.

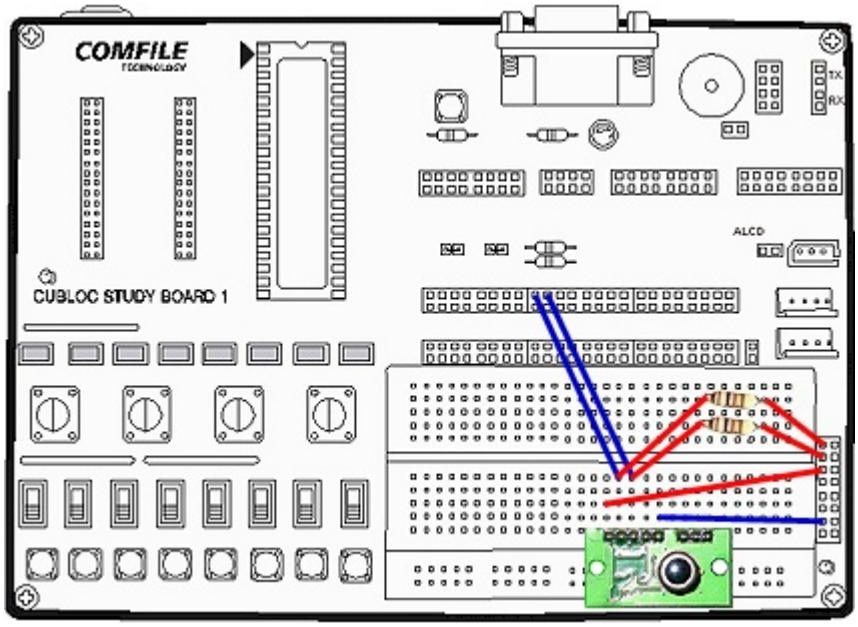
Notions abordées :
- Gestion I2C™

Matériel nécessaire :
- Une platine « CUBLOC Study Board » (facultative)
- 1 module « MTP81 »

Préparation matérielle :

Celle-ci repose sur le schéma théorique ci-dessous. Afin de faciliter la description de cette note d'application, nous utiliserons une platine « CUBLOC Study Board » associée à un module CUBLOC™ CB220 (voir schéma de raccordement ci-après).

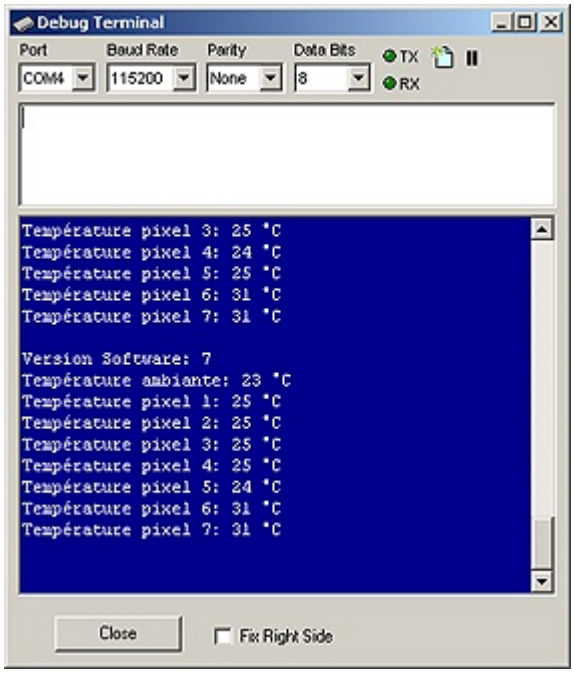




Les résistances ont pour valeur 4,7 Kohms.

Saisissez ensuite le petit programme présenté ci-dessus (ce dernier est disponible sur notre site Internet : www.lextronic.fr ou sur notre CD-ROM sous le nom « MTP81 »).

Ce programme va permettre de venir lire les registres du module « MTP81 ». Dès lors, vous pourrez voir apparaître dans la fenêtre de DEBUG le N° du firmware du module « MTP81 », la valeur de la température ambiante, puis celle de chaque « zone de sensibilité (pixel) » du détecteur (Dans l'exemple ci-dessous une main a été placée surle côté du capteur).



```
'###################################
'#        Gestion d'un module MTP81       #
'#      @Lextronic 2006 - 21/02/2006      #
'###################################

Const Device = CB220
Dim errorcom As Byte
Dim version As Byte
Dim temp As Byte
Dim pixel1 As Byte
Dim pixel2 As Byte
Dim pixel3 As Byte
Dim pixel4 As Byte
Dim pixel5 As Byte
Dim pixel6 As Byte
Dim pixel7 As Byte

Set I2c 8,9

Do
  I2cstart                                                    ' Condition Start I2C
  errorcom = I2cwrite (&HD0)        ' Adresse du module MSU08
  errorcom = I2cwrite (0)           ' Sélectionne l'adresse du premier registre à lire
  I2cstart                                                    ' Condition Start I2C
  errorcom = I2cwrite (&HD1)   ' Sélectionne condition de lecture I2C
  version = I2cread(0)                ' Récupere N° de révision du module MTP81
  errorcom = I2cwrite (&HD1)   ' Sélectionne condition de lecture I2C
  temp = I2cread(0)                       ' Récupere lecture température ambiante
  errorcom = I2cwrite (&HD1)   ' Sélectionne condition de lecture I2C
  pixel1=I2cread(0)                                           ' Récupère température pixel 1
  errorcom = I2cwrite (&HD1)   ' Sélectionne condition de lecture I2C
  pixel2=I2cread(0)                                           ' Récupère température pixel 2
  errorcom = I2cwrite (&HD1)   ' Sélectionne condition de lecture I2C
  pixel3=I2cread(0)                                           ' Récupère température pixel 3
  errorcom = I2cwrite (&HD1)   ' Sélectionne condition de lecture I2C
  pixel4=I2cread(0)                                           ' Récupère température pixel 4
  errorcom = I2cwrite (&HD1)   ' Sélectionne condition de lecture I2C
  pixel5=I2cread(0)                                           ' Récupère température pixel 5
  errorcom = I2cwrite (&HD1)   ' Sélectionne condition de lecture I2C
  pixel6=I2cread(0)                                           ' Récupère température pixel 6
  errorcom = I2cwrite (&HD1)   ' Sélectionne condition de lecture I2C
  pixel7=I2cread(0)                                           ' Récupère température pixel 7
 I2cstop                                                      ' Condition Stop I2C

  Debug "Version Software: ",Dec version,Cr
  Debug "Température ambiante: ",Dec temp," °C",Cr
  Debug "Température pixel 1: ",Dec pixel1," °C",Cr
  Debug "Température pixel 2: ",Dec pixel2," °C",Cr
  Debug "Température pixel 3: ",Dec pixel3," °C",Cr
  Debug "Température pixel 4: ",Dec pixel4," °C",Cr
  Debug "Température pixel 5: ",Dec pixel5," °C",Cr
  Debug "Température pixel 6: ",Dec pixel6," °C",Cr
  Debug "Température pixel 7: ",Dec pixel7," °C",cr,cr
  Delay 3000
Loop
```

NOTE D'APPLICATION # 10.
Mini jeu vidéo « CUBLOC ATTACK »

Cette note d'application va vous permettre de réaliser un petit jeu vidéo au moyen d'un simple afficheur LCD 4 x 20 caractères, de 3 boutons poussoir et d'un buzzer (si vous voulez du son). Nous l'avons appelé « CUBLOC ATTACK ». Il s'agit en fait d'une adaptation du très célèbre jeu des « envahisseurs ». Le but du jeu est simple : empêchez une horde de 5 envahisseurs d'atteindre le bas de l'écran en tirant dessus. Ces envahisseurs (aliens) se déplacent de gauche à droite et de droite à gauche en descendant d'un niveau lorsqu'ils atteignent une extrémité de l'écran LCD. Vous disposez d'une base (canon) au bas de l'écran que vous pouvez déplacer horizontalement à l'aide de 2 boutons-poussoirs. Un 3ème bouton-poussoir sert à envoyer des missiles. Vous ne pouvez tirer qu'un seul missile à la fois (c'est à dire qu'il ne vous sera pas possible de tirer un autre missile tant que le missile déjà tiré n'a pas atteint le haut de l'écran où qu'il n'a pas touché un alien). Dès que vous avez éliminé les 5 aliens de l'écran, vous passez au niveau suivant (avec des aliens plus rapides et de forme différente). Arriverez vous à battre le « high Score » ?

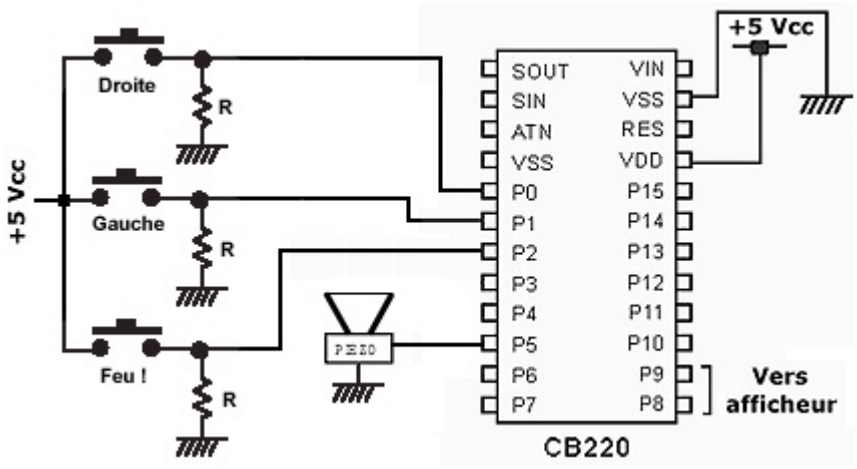
A chaque fois que vous éliminez un envahisseur, vous gagnez des points. Plus les envahisseurs sont touchés vers le haut de l'écran plus les points seront élevés. Le jeu mémorise le meilleur score réalisé (High Score) en l'affichant sur la bas de l'écran de présentation (appuyez sur le bouton de tir pour démarrer une nouvelle partie). Si ne serait-ce qu'un seul envahisseur arrive à atteindre le niveau de votre base…vous avez perdu.

Notions abordées :
- Gestion afficheur LCD
- Gestion de boutons-poussoir
- Génération de fréquence (pour l'émission des sons).

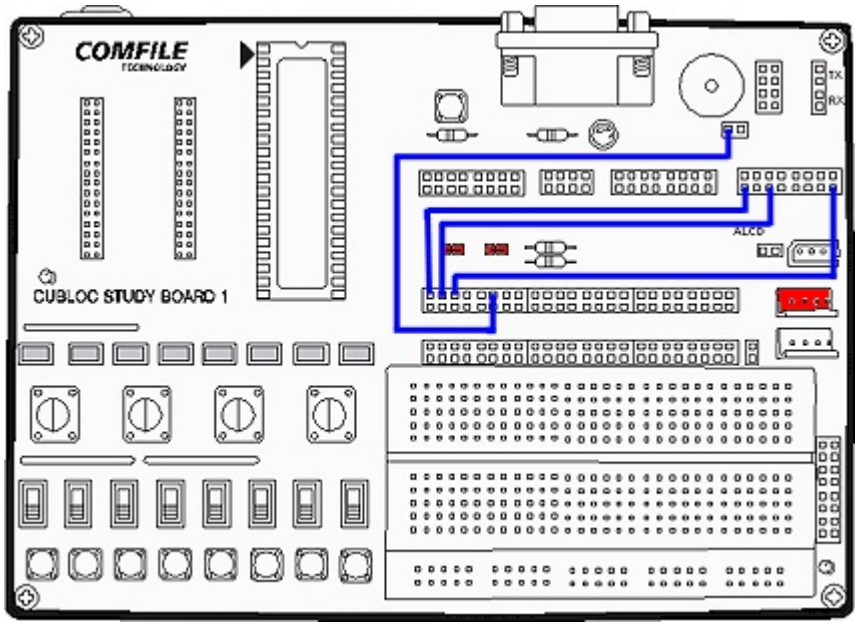
Matériel nécessaire :
- Une platine « CUBLOC Study Board » (facultative)
- 1 afficheur 4 x 20 caractères + 3 boutons-poussoirs + 1 buzzer sans oscillateur

Préparation matérielle :

Celle-ci repose sur le schéma théorique ci-dessous (les résistances ont pour valeur 10 K).



Afin de faciliter la description de cette note d'application, nous utiliserons une platine « CUBLOC Study Board » associée à un module CUBLOC™ CB220 (voir schéma de raccordement ci-après). Un afficheur 4 x 20 caractères devra être relié sur le port CuNET du CUBLOC à l'aide du connecteur prévu à cet effet.



Saisissez ensuite le petit programme présenté ci-dessus (ce dernier est disponible sur notre site Internet : www.lextronic.fr ou sur notre CD-ROM sous le nom « invaders »).

Interprétation du programme :
Le programme préssenté ci-après Peut aisément être optimisé (il est ainsi possible de limiter le nombre de variables utilisées). Toutefois ce dernier a volontairement été écrit de la sorte afin que vous puissiez facilement en comprendre le fonctionnement. Il vous sera dès lors possible de le modifier à volonté afin d'ajouter plus d'aliens, de modifier leur forme, leur vitesse, etc…

Le programme commence par l'affichage de l'écran de présentation (sur lequel apparaît le « high Score » (meilleur score). En appuyant sur la touche de tir, vous démarrez une partie. Dès lors le programme initialise les variables du jeu. La variable « mort » correspond au nombre d'aliens à tuer par partie (si vous ajoutez des aliens à l'écran, il vous faudra modifier la variable « mort » en conséquence). Le programme redéfini ensuite la forme de votre base et celle des aliens qui apparaîtrons sur l'afficheur (la forme des aliens est sélectionnée fonction de la valeur de la variable « aliens ». Consultez la note d'application du « bargraph de précision » présent dans cette note d'application pour plus d'information sur la redéfinition de caractères des afficheurs LCD).

Dans notre cas, nous avons défini 8 formes d'aliens différents qui reviennent cycliquement. Il vous sera facilement ainsi possible de modifier la forme de vos aliens dans ces lignes si vous le désirez. La position de chaque aliens et de votre base est mémorisée dans des variables (« base » et « vaisseaux » de type « tableau » à 20 éléments (correspondant aux 20 caractères d'une ligne de l'écran LCD). Avant de commencer une partie, le programme rempli les 20 éléments du tableau par des espaces (valeur 20) et ensuite la position des aliens par le caractère 10 (ce caractère correspond au caractère qui a été préalablement redéfini). La position de la base est associée au caractère 9 (également redéfini). Il est également très facilement possible de redéfinir plusieurs caractères à la fois et d'afficher ainsi des aliens de formes différentes à la fois.

Le principe général du programme repose sur la gestion en « temps » partagé des actions. Ainsi le déplacement de votre base sera géré en « temps » réel (c'est à dire que le déplacement de celle-ci se fera sans délai). Pour ce qui concerne le déplacement des aliens et de vos missiles, le programme a recours à l'utilisation de 2 « bases » de temps matérialisés par les variables « deplace » et « dep_miss ». Ces variables de type « single » sont incrémentées dans la boucle principale de 0,025 unités (pour les aliens) et de 0,010 pour vos tirs. Dès que ces variables dépassent la valeur 1 on autorise une action. Dans le cas des aliens, il s'agit d'autoriser leur déplacement d'une case. Dans le cas de votre tir, il s'agit du déplacement de votre missile d'une case vers le haut. Il est dés lors facilement possible de modifier ces « bases de temps » pour que les aliens et/ou vos tirs se déplacent plus vite ou moins vite.

Le programme permet également de rendre inopérant le bouton de tir si un missile est déjà présent à l'écran. Un test de « collision » entre votre missile et les aliens vérifie en fait la position x,y du missile avec celle des aliens. Si un aliens est présent, on remplace son caractère par un espace dans la variable du tableau « vaisseaux » afin qu'il s'apparaisse plus à l'écran. Une petite explosion est ensuite simulée par allumage successif des caractères « * # + » à l'emplacement de l'alien détruit.

Le reste du programme gère les déplacements des aliens (de gauche à droite et de droite à gauche avec un test de passage au niveau inférieur si un des aliens arrive sur un des bords et la fin de partie si les aliens atteigne la 4ème ligne de l'écran LCD).

Le programme prend également en charge la gestion des sons avec la génération de 2 fréquences alternées suivant les déplacements des aliens, la génération d'un petit « bip» lors d'un tir et d'un autre effet sonore lorsqu'un alien est touché.

```
'###################################
'#             CUBLOC ATTACK       #
'#    @Lextronic 2006 - 27/02/2006 #
'###################################

Const Device = CB220
Set Display 2,0,1,50                    ' Initialisation de l'afficheur LCD

Dim base(20) As Byte                    ' Mémoire emplacement de votre base
Dim vaisseaux(20) As Byte               ' Mémoire emplacement des aliens
Dim deplace As Single                   ' Base de temps déplacement aliens
Dim dep_miss As Single                  ' Base de temps déplacement missile
Dim niveau As Single                    ' Niveau (vitesse de déplacement des aliens)
Dim level As Byte                       ' Niveau du jeu
Dim a As Byte                           ' Variable d'utilisation générale
Dim sens As Byte                        ' Sens de déplacement aliens
Dim tir As Byte                         ' Gestion tir missile
Dim xmissile As Byte                    ' Position x du missile
Dim aliens As Byte                      ' Variable gestion forme des aliens
Dim affiche As String*20                ' Gestion affichage
Dim score As Integer                    ' Score
Dim highscore As Integer                ' Meilleur score
Dim mort As Byte                        ' Nombre aliens tués par niveau
Dim position As Byte                    ' Position x  de votre base
Dim attaque As Byte                     ' Position ligne des aliens
Dim son As Byte                         ' Type de son à jouer

Input 0                                 ' Configure les ports de conversion analogique/numérique en entrée
Input 1
Input 2
Low       5                             ' Port 5 en sortie
highscore = 0                           ' initialise valeur meilleur score


'************************ ecran de présentation ************************

present:
Cls
Delay 200
Csroff
Delay 200
score = 0                               ' Remise à zéro du score
niveau = 0.025                          ' RAZ niveau (vitesse aliens)
aliens = 0
level = 0
Locate 0,0
Print "** CUBLOC ATTACK **"
Locate 0,2
Print "High"
Locate 0,3
Print "Score: ",Dec highscore
Locate 13,3
Print "<Start>"
Do While Keyinh(2,20) = 0               ' Attend appui sur touche "feu"
Loop


'*************************** Nouvelle partie ***************************
Nouveau:
mort = 5                                ' Il y a 5 aliens à abattre
position = 10                           ' Position initiale de votre base
sens = 0                                ' Les aliens se déplacent vers la droite
attaque = 0                             ' Les aliens sont tout en haut
tir = 3                                 ' Il n'y a aucun missile de tiré
```

```
dep_miss = 0                                ' Initialise base de temps déplacement des missiles
deplace = 0                                 ' Initialise base de temps déplacement des aliens
son = 0                                     ' Initialise type de son à jouer
Cls
Delay 200
Csroff
Delay 200
Select Case aliens
  Case 0
    Print &H1B,&H44,10,27,17,31,21,14,14,17,27          ' Redéfinition caractères
  Case 1
    Print &H1B,&H44,10,14,14,21,14,14,10,17,10          ' Redéfinition caractères
  Case 2
    Print &H1B,&H44,10,4,14,21,21,14,14,21,21           ' Redéfinition caractères
  Case 3
    Print &H1B,&H44,10,4,14,21,21,14,14,21,21           ' Redéfinition caractères
  Case 4
    Print &H1B,&H44,10,4,14,21,14,14,21,21,21           ' Redéfinition caractères
  Case 5
    Print &H1B,&H44,10,27,14,21,21,31,17,14,27          ' Redéfinition caractères
  Case 6
    Print &H1B,&H44,10,4,14,21,31,31,21,14,4            ' Redéfinition caractères
  Case 7
    Print &H1B,&H44,10,17,17,31,21,14,14,17,17          ' Redéfinition caractères
End Select
aliens = aliens + 1
if aliens = 8 then aliens = 0
Print &H1B,&H44,9,4,4,14,14,31,31,31,31                 ' Redéfinition caractères de votre base
Delay 200


'********** initialisation des mémoires de position des aliens ********

For a = 0 To 19
  base(a)=20
  vaisseaux(a)=20
Next
base(10)=9                                              ' Caractère 9 redéfini pour la base
vaisseaux(0)=10
vaisseaux(1)=10
vaisseaux(2)=10
vaisseaux(3)=10
vaisseaux(4)=10

Do

'*************** Test si tous les aliens du niveau sont mort ? **********
If mort = 0 Then
  Locate xmissile,tir                                   ' Efface votre missile
  Print " "
  Freqout 0,0
  Locate 0,0
  level = level + 1                                     ' Niveau suivant
  Print "Niveau ",Dec level," termine"
  Locate 0,1
  Print "Score: ",Dec score
  Delay 3000
  Locate 0,1
  Print "                "
  niveau = niveau + 0.015                               ' Augmente vitesse déplacement des aliens
  Goto nouveau
End If
```

```
' ********** Raffraichissement de l'affichage **********

deplace = deplace + niveau
affiche = ""
For a = 0 To 19
   affiche = affiche + Chr(base(a))
Next
Locate 0,3
Print affiche                                          ' Affichage de votre base
affiche = ""
For a = 0 To 19
   affiche = affiche + Chr(vaisseaux(a))
Next
Locate 0,attaque
Print affiche                                          ' Affichage des aliens
If attaque = 3 Then Goto perdu


' ********** Test collision missile <> aliens **********
If tir <> 3 Then
  dep_miss=dep_miss+ 0.10
  Locate xmissile,tir                                  ' Affiche le missile
  Print "|"
  If tir = attaque And vaisseaux(xmissile)<>20 Then
    Locate xmissile,tir                                ' Affichage explosion
    Print "*"
    Freqout 0,9000
    Delay 70
    Locate xmissile,tir
    Print "#"
    Freqout 0,8000
    Delay 70
    Locate xmissile,tir
    Print "+"
    Freqout 0,7000
    Delay 70
    Locate xmissile,tir
    Print " "
    tir = 3                                            ' RAZ paramètres envoi missile
    dep_miss=0
    vaisseaux(xmissile)=20
    mort = mort -1
    score= score + (5-attaque)
  End If
End If


' ********** Gestion déplacement de votre base **********

If Keyinh(1,20) = 1 And position <>19 Then              ' Déplacement vers la droite
   base(position) = 20
   position = position + 1
   base(position) = 9                                  ' Caractère 9 redéfini pour la base
End If

If Keyinh(0,20) = 1 And position <>0 Then               ' Déplacement vers la gauche
     base(position) = 20
     position = position - 1
     base(position) = 9                                ' Caractère 9 redéfini pour la base
End If

If Keyinh(2,20) = 1 And tir = 3 Then                    ' Tir d'un missile
     Freqout 0,400
     xmissile=position
```

```
      tir = 2
      Locate xmissile,tir
      Print "|"
      Gosub gson
    End If


   ' ************** Gestion déplacement des aliens vers la droite **************

  If deplace > 1 And sens = 0 Then                ' Gestion déplacement vers la droite
    Gosub gson                                    ' Test gestion du son
    If vaisseaux(19) <> 20 Then                   ' Test si les aliens sont arrivés complètement à droite ?
       Gosub efface                               ' Efface les aliens de la ligne courante
       sens=1                                     ' Et ils repartent vers la gauche
    Else
       For a = 19 To 1 Step -1                    ' Tous les aliens se déplacent vers la droite
          vaisseaux(a) = vaisseaux(a-1)
       Next
      vaisseaux(0)=20                             ' On efface la première position à gauche
    End If
    deplace = 0
End If


' *********** Gestion déplacement des aliens vers la gauche *****************

If deplace > 1 And sens = 1 Then                  ' Gestion déplacement vers la gauche
   Gosub gson                                     ' Test gestion du son
   If vaisseaux(0) <> 20 Then                     ' Test si les aliens sont arrivés complètement à gauche ?
      Gosub efface                                ' Efface les aliens de la ligne courante
      sens=0                                      ' Et ils repartent vers la droite
   Else
      For a = 0 To 18                             ' Tous les aliens se déplacent vers la droite
        vaisseaux(a) = vaisseaux(a+1)
      Next
      vaisseaux(19)=20                            ' On efface la première position à gauche
  End If
  deplace = 0
End If


    ' ******************* Gestion déplacement des missiles ************************

If dep_miss>1 And tir <>3 Then
   If tir = 0 Then
      Locate xmissile,tir
      Print " "
      tir =3
      dep_miss=0
    Else
       Locate xmissile,tir
       Print " "
       tir = tir -1
       Locate xmissile,tir
       Print "|"
       dep_miss=0
    End If
End If
Loop


' ******************** Routine de gestion d'affichage ************************

  perdu:
```

```
  Locate xmissile,tir                                    ' Efface votre missile
  Print " "
  Freqout 0,0
  Locate 0,0
  Print "Vous avez perdu..."
  Locate 0,1
  Print "Score: ",Dec score
  Locate 0,2
  Print "Niveau: ",Dec level
  Delay 4500
  If score > highscore Then highscore = score
 Goto present


 ' ******************** Routine de gestion d'affichage ************************

efface:
  Locate 0,attaque
  Print String(20,20)
  attaque = attaque + 1                                  ' Les aliens descendent d'une ligne !!!
  Return


 ' ******************** Routine de gestion des sont ************************
gson:
    If son = 0 Then
      Freqout 0,40000
      son = 1
    Else
      Freqout 0,48000
      son = 0
    End If
    Return
```

NOTE D'APPLICATION # 11.
Gestion d'un modem radio « TDL2A-433-9 »

Cette note d'application va vous permettre de piloter un module radio « TDL2A-433-9 » du fabricant Radiometrix™. Ce dernier est un modem radio subminiature low cost synthétisé pouvant facilement s'intégrer au sein de nouveaux projets ou d'applications existantes. Il assurera la transmission à distance de signaux numériques séries de façon totalement transparente (le module génère les trames de préambule, de synchro, ainsi que la mise en "paquet" et le codage Manchester des données tout en effectuant un checksum). D'un point de vue utilisation celui ci s'apparente en fait à un simple câble série « virtuel » très rapide et simple à mettre en œuvre.

Notions abordées :
- Gestion communications séries.

Matériel nécessaire :
- Une platine « CUBLOC Study Board » (facultative)
- 2 modules « TDL2A-433-9 » (ou un module « TDL2A-433-9 » et un module « TDL2i »
- 4 leds + résistances.

Cette note d'application vous montera comment interfacer ce modem avec un CB220 afin de pouvoir réaliser une télécommande radio 4 canaux bidirectionnelle (via un compatible PC) avec fonction d'accusé de réception des ordres. Le module « TDL2A-433-9 » dispose d'une interface série (niveau logique 0 / + 5V qui pourra très facilement s'interfacer directement avec le CUBLOC). Du côté du PC, il vous faudra ajouter une circuit de mise à niveau MAX-232 afin d'adapter les signaux +/-12 V du port RS-232 avec les niveaux du « TDL2A-433-9 » (vous trouverez un schéma d'application « type » sur le site du fabricant : www.radiometrix.co.uk ou www.radiometrix.com).

Il est également possible d'avoir recours à un module « TDL2i », lequel intègre sur une même platine le modem radio, une prise SUB-D 9 broches, son antenne et le circuit d'interfaçage Max-232 (de telle sorte qu'il vous suffira simplement de le relier au port RS-232 du PC pour le rendre opérationnel). Voir plus d'infos ici : http://www.radiometrix.co.uk/products/tdl2i.htm

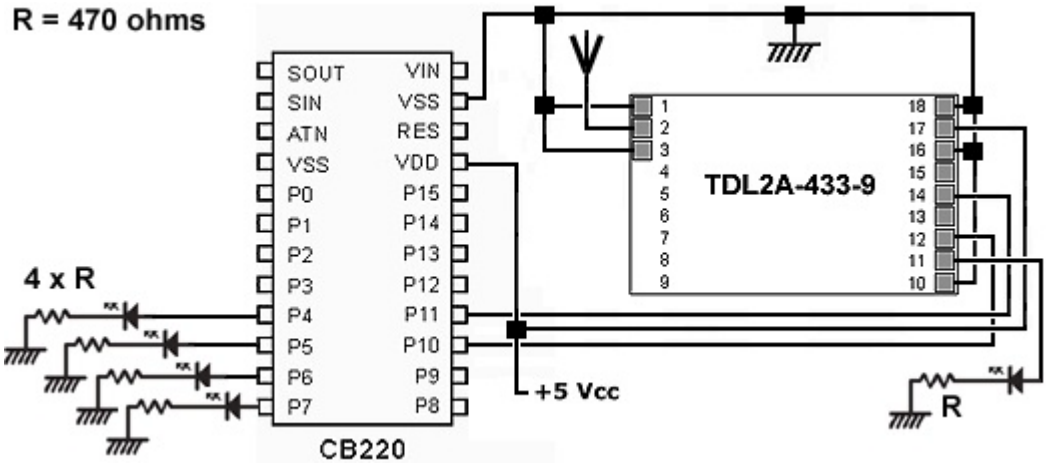
Préparation matérielle :

Du côté du PC, celle-ci consiste à relier la platine « TDL2i » à un port COM de libre du PC et à exécuter le programme de communication « HyperTerminal™ ». Ce programme se trouve via les menus :

Démarrer -> **Programmes** -> **Accesoires** -> **Communications** -> **HyperTerminal.**

Une fois le programme exécuté, donnez un nom à la communication (par exemple CUBLOC), puis sélectionnez le N° de port sur lequel est connecté le module « TDL2i » dans l'onglet du bas de la fenêtre. Validez par OK. Sélectionnez enfin une vitesse de connexion de 9600 bds / 8 bits / Parité (aucun) / 1 bit de stop / Contrôle de flux (aucun) et validez à nouveau. Puis alimentez le module « TDL2i » à l'aide d'une pile 9 V.

Du côté du module CUBLOC, il vous faudra réaliser le montage ci-dessous.



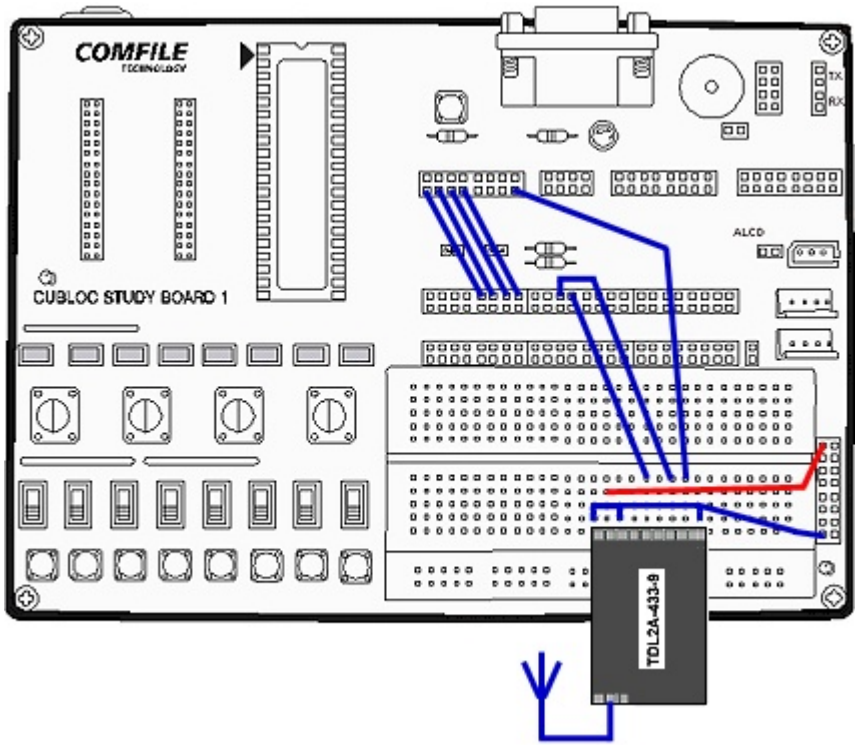
Principe de fonctionnement de l'application :

Le programme HyperTerminal™ présent sur le PC vous permettra d'envoyer des ordres séries depuis le clavier de votre PC, lesquels seront transmis au module CUBLOC via les 2 modems radio. Ainsi en sollicitant les touches 1 à 4 du PC, vous changerez alternativement l'état des 4 sorties du CUBLOC (une sollicitation de la touche 1 du PC allume la Led de la première sortie du CUBLOC – une seconde sollicitation de la touche 1 du PC éteint la Led de la première sortie et ainsi de suite pour les touches 2 à 4 et les sorties Leds 2 à 4 du module CUBLOC).

Le programme HyperTerminal™ peut également recevoir des données en provenance du port sortie en les affichant à l'écran. La note d'application utilise cette possibilité pour renvoyer après chaque commande l'état des 4 sorties du CUBLOC. Ainsi après avoir sollicité une des touches 1 à 4 au niveau du PC, le CUBLOC change l'état de la sortie adéquat et renvoie sous forme de textes l'état des 4 sorties qui s'affichera sur la fenêtre du PC (vous serez ainsi sûr que l'ordre radio a correctement été reçu et que la sortie a bien l'état que vous vouliez lui faire prendre).

En sollicitant la touche 0 du PC, le CUBLOC ne modifie aucun état de ses sorties, mais renvoie tout de même l'état de ces dernières. La touche 0 fait ainsi office de touche d'interrogation en quelque sorte (dans le cas de figure où vous avez un doute sur l'état des sorties du module CUBLOC).

Réalisation du montage à l'aide de la platine « Cubloc Study Board » :



Afin de faciliter la mise en œuvre du modem radio côté CUBLOC, il vous sera possible d'utiliser une platine «CUBLOC Study Board» comme le montre le schéma ci-dessus. Le module «TDL2A-433-9» sera monté « en l'air » et implanté d'un seul côté sur la plaque de connexion sans soudure. Le raccordement des broches « en l'air » du modem pourra se faire au moyen d'un connecteur avec fils au pas de 2.54 mm (Réf. BTWF3). vous pourrez ainsi utiliser un de ces fils pour réaliser l'antenne du modem). A noter qu'une des Leds de la platine est reliée sur la sortie « STATUS » du Modem afin de pouvoir vérifier que ce dernier réceptionne correctement les trames radios.

Interprétation du programme : (Ce dernier est présent sur notre CD sous le nom « Tdl2A »)
Le programme commence par l'initialisation des variables représentant l'état des 4 sorties du module CUBLOC (et de leur initialisation « physique » au niveau logique 0 V). Le programme continu ensuite par l'analyse du contenu du buffer série de réception du module CUBLOC. Si le CUBLOC a reçu un code ASCII correspondant à une des touches 1 à 4 du PC, il s'en suit un changement d'état intermittent (via une fonction XOR) des sorties en fonction du caractère reçu dans le buffer. La valeur 49 correspond à la valeur ASCII du chiffre de la touche 1 sollicitée sur le PC. La valeur 50 correspond au chiffre 2, etc…Une fois la variable de la sortie remise à jour, le programme force la variable de réception avec la valeur 48 (correspondant à la valeur ASCII du chiffre de la touche 0 du clavier – ceci afin de « forcer » le reste du programme à renvoyer un accusé de réception). Le programme remet ensuite à jour l'état « physique » des sorties puis envoi un accusé de réception de l'état de chaque sortie via des phrases dont chaque lettre est transmise au module radio « TDL2A-433-9 » afin que ces phrases s'affichent dans la fenêtre de HyperTerminal™ du PC.

NOTE IMPORTANTE :
Il est important de signaler que le programme de cette note d'application ne fait office que de programme d'évaluation et de test et qu'il ne devra en aucun cas être utilisé tel quel au sein d'une application. En effet, les sorties du CUBLOC sont directement activées via la réception d'un seul octet. Dès lors n'importe quel autre module « TDL2A-433-9 » émettant à proximité de votre application pourra être susceptible de déclencher vos sorties si ce dernier envoi un code ASCII correspondant aux touches 1 à 4. Dans le cadre d'une application réelle, il vous faudra impérativement envoyer au préalable un code sur plusieurs octets correspondant à une sorte de « mot » de passe ou de « clef de sécurité » (ce code pourra être répété avant et après les ordres de commande). Dès lors le CUBLOC devra avant toute activation de ses sorties s'assurer de la validité de ce code afin que vous soyez sûr qu'il s'agisse bien de votre application qui essai de vous envoyer un ordre radio.

```
'####################################
'#        Gestion d'un modem radio          #
'#       Radiometrix "TDL2A-433-9"          #
'#     @Lextronic 2006 - 24/02/2006         #
'####################################

Const Device = CB220

Dim sortie1 As Byte
Dim sortie2 As Byte
Dim sortie3 As Byte
Dim sortie4 As Byte
Dim reception As Byte

Opencom 1,9600,3,10,35

'****************** RAZ sorties *****************
Low 4
```

```
Low 5
Low 6
Low 7
Sortie1 = 0
Sortie2 = 0
Sortie3 = 0
Sortie4 = 0

'****************** Boucle principale *************
Do
   reception=Get(1,1)
   Select Case reception
     Case 49
       sortie1=sortie1 Xor &HFF
       reception=48
     Case 50
       sortie2=sortie2 Xor &HFF
       reception=48
     Case 51
       sortie3=sortie3 Xor &HFF
       reception=48
     Case 52
       sortie4=sortie4 Xor &HFF
       reception=48
   End Select

 ' ********** Mise à jour état des sorties *********
   delay 100
   Out 4,sortie1
   Out 5,sortie2
   Out 6,sortie3
   Out 7,sortie4

   If reception = 48 Then
     If sortie1=0 Then
       Putstr 1,"La sortie N° 1 est non active.",13
     Else
       Putstr 1,"La sortie N° 1 est active.",13
     End If
     Delay 100
     If sortie2=0 Then
       Putstr 1,"La sortie N° 2 est non active.",13
     Else
       Putstr 1,"La sortie N° 2 est active.",13
     End If
     Delay 100
     If sortie3=0 Then
       Putstr 1,"La sortie N° 3 est non active.",13
     Else
       Putstr 1,"La sortie N° 3 est active.",13
     End If
     Delay 100
     If sortie4=0 Then
       Putstr 1,"La sortie N° 4 est non active.",13,13
     Else
       Putstr 1,"La sortie N° 4 est active.",13,13
     End If
    End If
   Loop
```

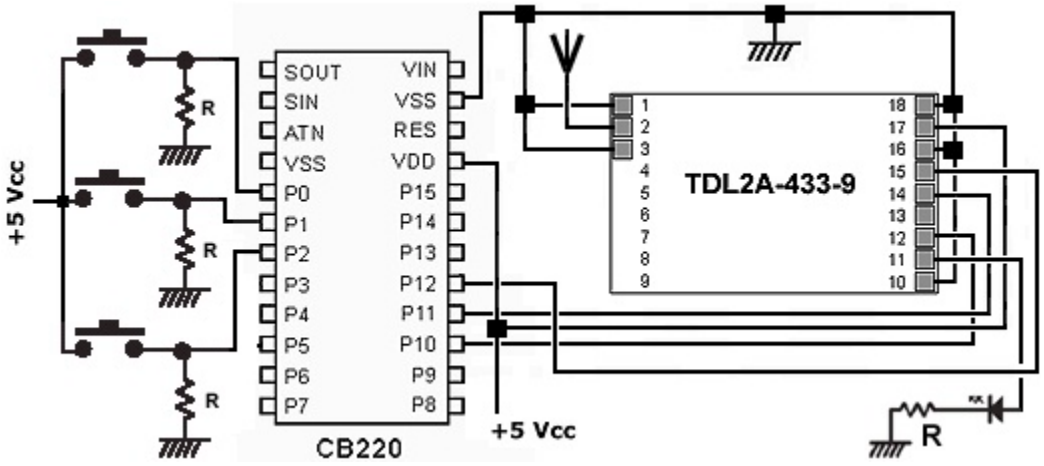
NOTE D'APPLICATION # 12.
Configuration du modem radio « TDL2A-433-9 »

Cette note d'application va vous permettre de reconfigurer certains paramètres du modem radio « TDL2A-433-9 » à l'aide du module CB220. En effet, chaque modem dispose d'un système d'adressage interne paramétrable qui vous permettra de concevoir des mini-réseaux de communication (chaque module peut ainsi être paramétré sur 8 adresses différentes et seuls les modules présentant la même adresse pourront dialoguer ensemble). Il est également possible de sélectionner 5 fréquences de travail pré-définies dans la bande 433 MHz afin que plusieurs couples de "TDL2A-433-9" puissent travailler dans la mesure du possible simultanément sur le même site (ou afin que vous puissiez modifier la fréquence des modules si un des canal est perturbé).

Notions abordées :
- Gestion communications séries et reconfiguration du modem radio « TDL2A-433-9 »

Matériel nécessaire :
- Une platine « CUBLOC Study Board » (facultative)
- 2 modules « TDL2A-433-9 » (ou un module « TDL2A-433-9 » et un module « TDL2i »
- 3 boutons-poussoirs

Cette note d'application utilise la même configuration que la note d'application n° 11. A savoir un modem « TDL2A-433-9 » raccordé au module CB220 et une platine « TDL2i » raccordée à un PC. Côté CB220, vous devrez réaliser le montage ci-dessous.



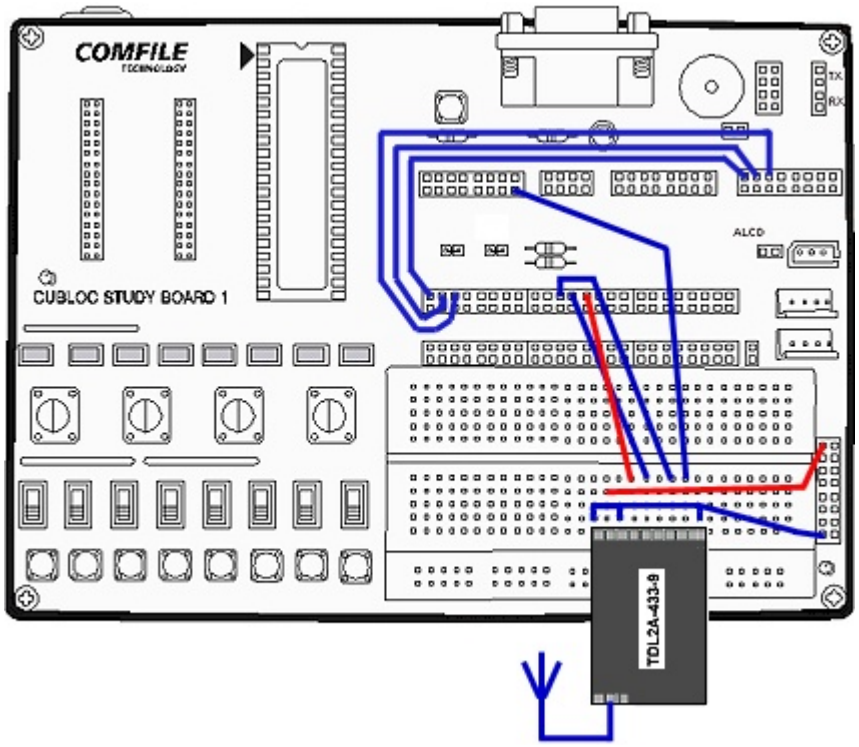
- Le premier bouton servira à modifier l'adresse du module. A chaque sollicitation, vous modifierez cette adresse et vous ferez afficher dans la fenêtre de debug du PC la valeur des paramètres de configuration du « TDL2A-433-9 » (ADDR et CHAN : que sont respectivement l'adresse du module et le canal utilisé).

- Le premier bouton servira à modifier le canal (la fréquence) du module. A chaque sollicitation, vous modifierez ce canal et vous ferez afficher dans la fenêtre de debug du PC la valeur des paramètres de configuration du « TDL2A-433-9 » (ADDR et CHAN : que sont respectivement l'adresse du module et le canal utilisé).

- Le 3ème bouton-poussoir de la platine sert à envoyer un petit texte pré-défini depuis le CB220 vers le PC par l'intermédiaire des modems. Ceci vous permettra de tester (si le texte s'afiche correctement) que les 2 modems sont bien sur la même adresse ET sur la même fréquence.

Le port P12 du CUBLOC est relié à la borne SETUP du module « TDL2A-433-9 » afin que le CB220 puisse activer son mode configuration lors de la sollicitation des boutons-poussoir 1 et 2.

Le but de cette note d'application sera d'apprendre à modifier successivement l'adresse et la fréquence des modems « TDL2A-433-9 » (via le CUBLOC CB220 et via le PC). Afin de faciliter la description, nous utiliserons une platine « CUBLOC Study Board ».



Saisissez ensuite le petit programme présenté ci-après (ce dernier est disponible sur notre site Internet : www.lextronic.fr ou sur notre CD-ROM sous le nom « Tdl2A2 »).

Interprétation du programme :
Côté PC, commencez par exécuter le programme « HyperTerminal™ » (voir la note d'application précédente pour plus d'info). Sollicitez ensuite le bouton-poussoir N°3 de la platine du CUBLOC. A ce stade, si tout fonctionne correctement un message envoyé par le CUBLOC via les modems radio doit s'afficher sur la fenêtre d'HyperTerminal™ du PC (la communication entre les 2 modules « passe » bien car ils disposent tous les 2 d'une adresse et d'une fréquence de travail identique).

Sollicitez alors le bouton-poussoir N° 1 de la platine du CUBLOC. A ce stade, un message apparaît dans la fenêtre de Debug du PC servant à programmer votre module CUBLOC. Dans ce dernier, on peu voir que le paramètre ADDR passe à 1. A ce moment, la broche SETUP du modem est mise à la masse via le port P12 du CUBLOC, lequel va ensuite envoyer les caractères ADDR1, suivi d'un retour chariot (caractère 13) et d'un LF (Line Feed – caractère 10). En fin d'envoi, le CUBLOC désactive le mode configuration du modem en faisant repasser l'entrée SETUP au niveau logique haut. A noter qu'entre chaque envoi de caractère, le CUBLOC réalise une petite temporisation. Cette tempo est nécessaire car le Modem renvoi systématiquement (lorsqu'il est en mode configuration) un écho des caractères qu'il reçoit. La durée de la tempo pourra être réduite (voir idéalement remplacée par une routine qui attendra la réception du caractère écho avant de passer à la suite de l'envoi).

A ce stade, sollicitez à nouveau le bouton-poussoir N°3 de la platine du CUBLOC. Dans ce cas, aucun message ne doit s'afficher dans la fenêtre d'HyperTerminal™ ce qui est normal car le modem du CUBLOC est sur l'adresse 1 (ADDR1), tandis que celui du PC est sur l'adresse ADDR0. Pour rétablir la communication, mettez le strap PGM du module « TDL2i » relié au port COM du PC (celui sur lequel vous avez HyperTerminal™). Puis tapez les lettres (en majuscule) ADDR1 (puis la touche Enter – puis restez appuyez sur la touche ALT et tout en restant appuyé, saisissez le nombre 010). Retirez le strap PGM du module « TDL2i » et appuyez à nouveau sur le bouton-poussoir N°3 de la platine du CUBLOC. Maintenant le message doit bien s'afficher sur l' écran d'HyperTerminal™ (puisque les 2 modules on la même adresse).

Le bouton-poussoir N°2 de la platine de CUBLOC vous permettra de la même façon de modifier la fréquence (le canal) du MODEM relié au CUBLOC. Et de la même façon, il vous suffira de saisir (après avoir remis le strap PGM sur le « TDL2i ») la commande CHANx (puis la touche Enter – puis restez appuyez sur la touche ALT et tout en restant appuyé, saisissez le nombre 010). X devra être identique à la valeur utilisée sur le MODEM associé au CUBLOC pour retrouver la communication entre les 2 systèmes.

La reprogrammation des paramètres du modem est accessible depuis 2 fonctions. Ces dernières peuvent être simplifiée (mais nous les avons délibérément écrites de la sorte afin que vous puissiez bien analyser la suite de caractères à envoyer au module pour le configurer).

Avec la possibilité de pouvoir modifier les adresses des modules, il vous sera possible de créer très facilement des mini réseaux de communication via l'utilisation de plusieurs modems sur un même site, en modifiant « à la volée » les paramètres ADDR et CHAN afin qu'un module « TDL2A-433-9 » puisse dialoguer avec un autre modem particulier (sans que les autres modems présents dans l'aire de couverture ne reçoivent les informations).

Avec la possibilité de pouvoir utiliser différentes fréquences, vous pourrez faire travailler plusieurs « couples » de modems sur le même site sans interférence.

A noter que la modification des paramètres ADDR et CHAN se fait uniquement dans la mémoire RAM du modem radio (dès lors en coupant leur alimentation et en les mettant à nouveau sous

tension, vous retrouvez les paramètres « usines » : ADDR0 et CHAN0). Il vous est toutefois possible d'enregistrer les paramètres modifiés dans la mémoire EEprom du modem radio à l'aide de la commande SETPROGRAM (Touche Enter). Dès lors lors d'une nouvelle mise sous tension, le module réutilisera les valeurs des paramètres sauvegardée. Attention l'enregistrement des paramètres en mémoire EEprom pourra nécessiter d'appliquer une temporisation légèrement plus important après la touche Enter. L'utilisation de la commande SETPROGRAM n'est pas montrée dans la note d'application.

```
'###################################
'#     Configuration d'un modem radio      #
'#       Radiometrix "TDL2A-433-9"         #
'#     @Lextronic 2006 - 24/02/2006        #
'###################################

Const Device = CB220

Dim v1 As Byte
Dim v2 As Byte
Dim a As Byte

Opencom 1,9600,3,5,50

'******************** RAZ port  ******************
Input 0                                                 ' Configure les ports en entrée
Input 1
Input 2
High  12                                                ' Configure le port en sortie
v1 = 0
v2 = 0
Delay 1000
Debug "ADDR",Dec v1,Cr,"CHAN",Dec v2,Cr,Cr

'****************** Boucle principale *************

Do
  If Keyinh(0,40) = 1 Then                              ' Test bouton de modification de l'adresse
    v1 = v1 + 1
    If v1 = 8 Then v1 = 0
      Debug "ADDR",Dec v1,Cr,"CHAN",Dec v2,Cr,Cr
      Do While Keyinh(0,40) = 1       ' Attend relachement de la touche
      Loop
      addr v1                                           ' Configure l'adresse du modem
    End If

    If Keyinh(1,40) = 1 Then                            ' Test bouton de modification du canal radio
      v2 = v2 + 1
      If v2 = 5 Then v2 = 0
        Debug "ADDR",Dec v1,Cr,"CHAN",Dec v2,Cr,Cr
        Do While Keyinh(1,40) = 1       ' Attend relachement de la touche
      Loop
      chan v2                                           ' Configure le canal (fréquence) du module
    End If

    If Keyinh(2,40) = 1 Then
       Putstr 1,"Si vous pouvez voir ce message sur l ecran du PC",13
       Delay 300
       Putstr 1,"C est que les 2 modules sont bien configures",13,13
       Delay 100
```

```
      Do While Keyinh(2,40) = 1                                    ' Attend relachement de la touche
      Loop
    End If
  Loop
  End


'#########################
'#   Sous Routine addr          #
'#########################

Sub addr(v3 As Byte)
  Low 12                                          ' Place la broche SETUP du TDL2A-433-9 au niveau bas
  Delay 100
  Putstr 1,"A"
  Delay 100
  Putstr 1,"D"
  Delay 100
  Putstr 1,"D"
  Delay 100
  Putstr 1,"R"
  Delay 100
  Putstr 1,v3+48
  Delay 100
  Putstr 1,13
  Delay 100
  Putstr 1,10
  Delay 100
  High  12                                       ' Place la broche SETUP du TDL2A-433-9 au niveau haut
  Delay 100
 End Sub


'#########################
'#   Sous Routine chan          #
'#########################

Sub chan(v3 As Byte)
  Low 12                                       ' Place la broche SETUP du TDL2A-433-9 au niveau bas
  Delay 100
  Putstr 1,"C"
  Delay 100
  Putstr 1,"H"
  Delay 100
  Putstr 1,"A"
  Delay 100
  Putstr 1,"N"
  Delay 100
  Putstr 1,v3+48
  Delay 100
  Putstr 1,13
  Delay 100
  Putstr 1,10
  Delay 100
  High  12                                          ' Place la broche SETUP du TDL2A-433-9 au niveau haut
  Delay 100
End Sub
```

NOTE D'APPLICATION # 13.
Gestion d'un module « ezKEY »

Cette note d'application va vous permettre de piloter un module « ezKEY ». Doté d'un buffer de 40 caractères, ce dernier vous permettra de récupérer les informations d'un clavier compatible PS/2 sous la forme de données séries (interfaçage simple au niveau TTL / 9600 bds via 2 fils) sans que vous ayez à maîtriser ou à connaître leur protocole et mode de fonctionnement.

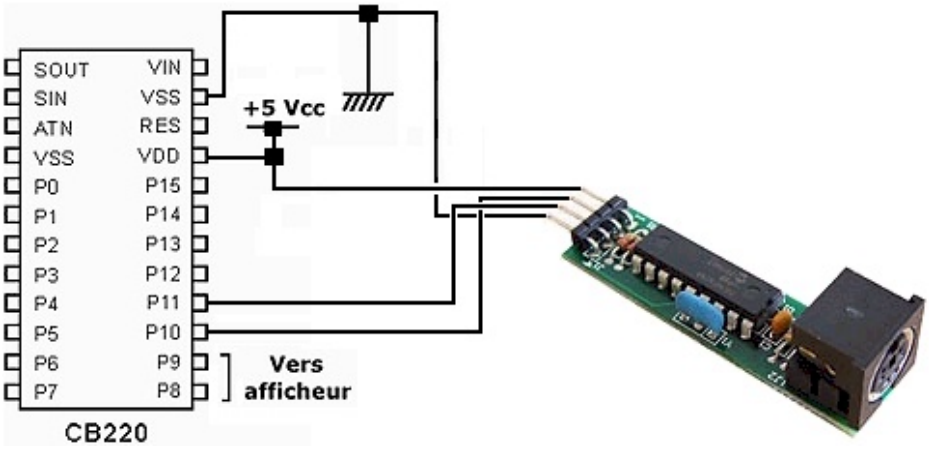
Pour cette application nous transformererons un écran LCD 4 x 20 caractères Comfile (relié sur le port CuNET) en petit éditeur de texte. Il vous sera ainsi possible de déplacer le curseur sur l'afficheur à l'aide des touches de directions du clavier, de saisir du texte, de gérer la touche « effacement » retour en arrière, ainsi que la touche « Enter ». On utilisera également la touche « ESC » pour effacer le contenu de l'afficheur et les touches « F1 » et « F2 » pour respectivement activer et désactiver le rétro éclairage de l'écran LCD. Lors des déplacement du curseur, des tests de déplacements sont effectués afin que la saisie reste confinée à l'intérieur de l'écran LCD.

Notions abordées :
- Gestion communications séries

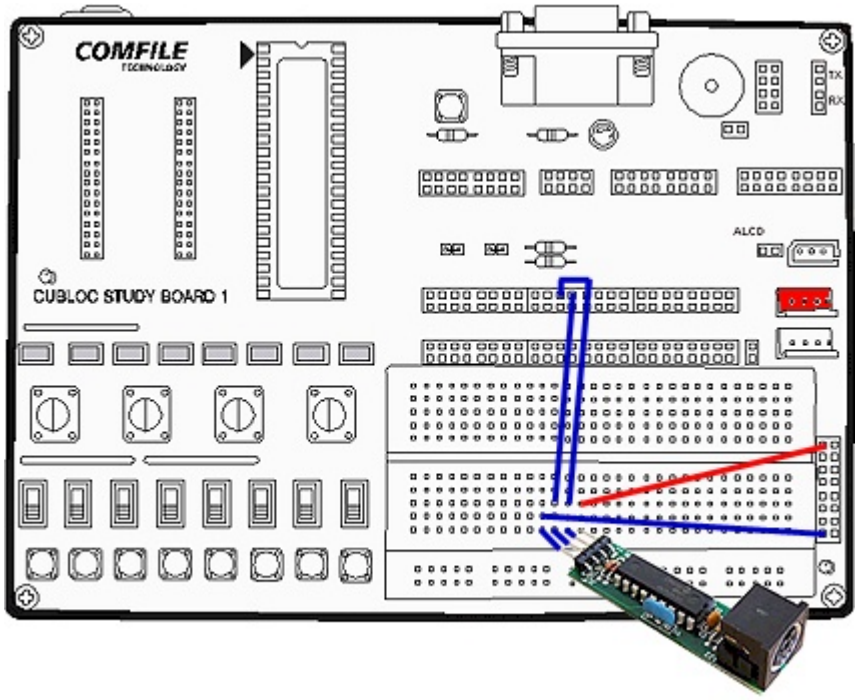
Matériel nécessaire :
- Une platine « CUBLOC Study Board » (facultative)
- 1 module « ezKEY » + 1 clavier de PC + 1 afficheur LCD 4 x 20 caractères

Préparation matérielle :

Celle-ci repose sur le schéma théorique ci-dessous.



Afin de faciliter la description de cette note d'application, nous utiliserons une platine « CUBLOC Study Board » associée à un module CUBLOC™ CB220 (voir schéma de raccordement ci-après). Un afficheur 4 x 20 caractères devra être relié sur le port CuNET du CUBLOC à l'aide du connecteur prévu à cet effet.



Saisissez ensuite le petit programme présenté ci-après (ce dernier est disponible sur notre site Internet : www.lextronic.fr ou sur notre CD-ROM sous le nom « ezkey »).

Interprétation du programme :
Ce dernier est très simple à comprendre. Le programme entame (après l'initialisation de la communication CuNET de l'afficheur LCD et du port série du CUBLOC) une temporisation de quelques secondes pour être certain qu'après la mise sous tension de l'ensemble, l'initialisation automatique du clavier soit terminée. Une seconde initialisation forcée (au cas ou) est ensuite réalisée via l'envoi du caractère « I » au module « ezKEY ». S'en suit alors une boucle sans fin dans laquelle on vient lire l'état du buffer de réception du module « ezKEY ». Si ce dernier n'est pas nul (si on a donc utilisé le clavier) on effectue alors une action en rapport avec la touche sollicitée (via l'instruction « select case »).

A noter que le module « ezKEY » décode les informations du clavier comme un clavier « QUERTY ». C'est à dire que pour la touche « a », vous récupérerez le caractère « q » est ainsi de suite. Il vous sera très facilement possible à l'aide de l'instruction « select case » de ré attribuer les bons caractères par rapport aux touches.

```
'##################################
'#    Gestion d'un module « ezKEY »      #
'#    @Lextronic 2006 - 24/02/2006       #
'##################################

Const Device = CB220

Dim mode As Byte
Dim a As Byte
Dim y As Byte
Dim x As Byte

Set Display 2,0,1,50                    ' Initialisation port I2C pour LCD
Opencom 1,9600,3,30,20                  ' Initialisation pour série

Delay 3000                              ' Attend l'initialisation du clavier
Cls                                     ' Effacement de l'écran
Delay 200
x = 0                                   ' initialisation position curseur du LCD
y = 0
a = 0                                   ' Initialise variable de réception

Putstr 1,"I"                            ' Initialise le clavier
Delay 3000

Bclr 1,0                                ' Efface buffer réception

Do
   Do   While a = 0
     Putstr 1,"G"                       ' Lance une commande d'interrogation du module ezKEY
     Delay 40
     a=Get(1,1)
    Loop

    Select Case a                       ' Test le caractère reçu ?
     Case 155                           ' Flèche du haut
       If y = 0 Then
         y = 3
       Else
         y = y -1
       End If
       Locate x,y
     Case 157                           ' Flèche du bas
       y = y + 1
       If y = 4 Then y = 0
       Locate x,y
     Case 158                           ' Flèche de droite
       x = x + 1
       If x = 20 Then x = 0
       Locate x,y
     Case 156                           ' Flèche de gauche
        If x = 0 Then
         x = 19
       Else
         x = x -1
       End If
       Locate x,y
     Case 27                            ' Touche "escape"
       Cls
       Delay 200
       x = 0
       y = 0
```

```
    Locate x,y
  Case 8                          ' Touche "efface"
    Locate x,y
    Print " "
    If x <> 0 Then x = x - 1
    Locate x,y
  Case 13                         ' Touche "Enter"
    x = 0
    y = y + 1
    If y = 4 Then y = 0
    Locate x,y
  Case 135                        ' Touche F1
    Print &H1B,&H42               ' Rétro-éclairage ON
  Case 136
    Print &H1B,&H62               ' Rétro-éclairage OFF
  Case Else                       ' Autres caractères
     Locate x,y
          Print a
          x = x + 1
          If x = 20 Then
           x = 0
           y = y + 1
           If y= 4 Then y = 0
           Locate x,y
           End If
               End Select
        a = 0
Loop
```

NOTE D'APPLICATION # 14.
Gestion d'un module RFID « UM005 »

Cette note d'application va vous permettre d'expérimenter la technologie « RFID » au moyen d'un petit module hybride spécialisé « UM005 ». Ce dernier ne nécessite qu'une simple antenne pour pouvoir effectuer la lecture de tag RFID de type 125 KHz Unique™. Il dispose d'une sortie Led, d'une sortie buzzer et d'une sortie série (format RS-232 aux niveaux logiques TTL). Cette sortie sera utilisée par le CUBLOC pour récupérer les informations inscrites sur les TAG RFID.

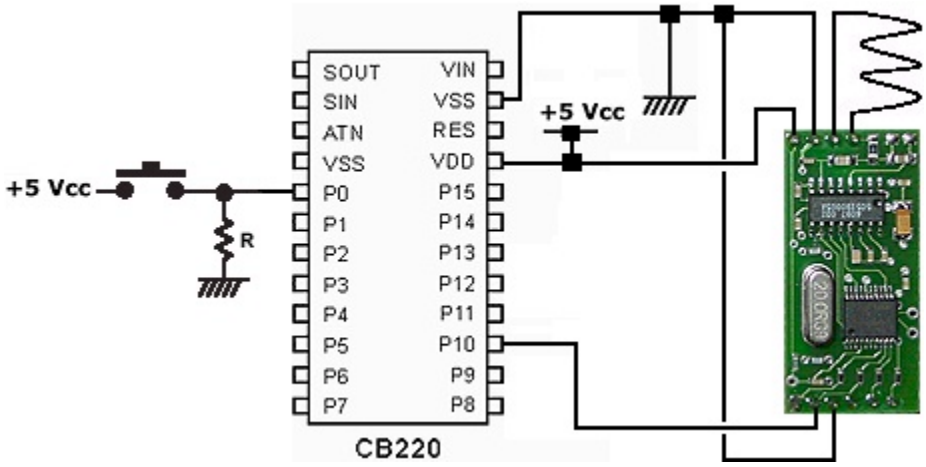
Il existe une multitude de tag (format carte bancaire, jeton, pastille, ampoule de verre…) lesquels disposent d'un code unique pré-enregistré en usine.

Notions abordées :
- Gestion d'une liaison série + enregistrement / lecture de données en mémoire EEprom

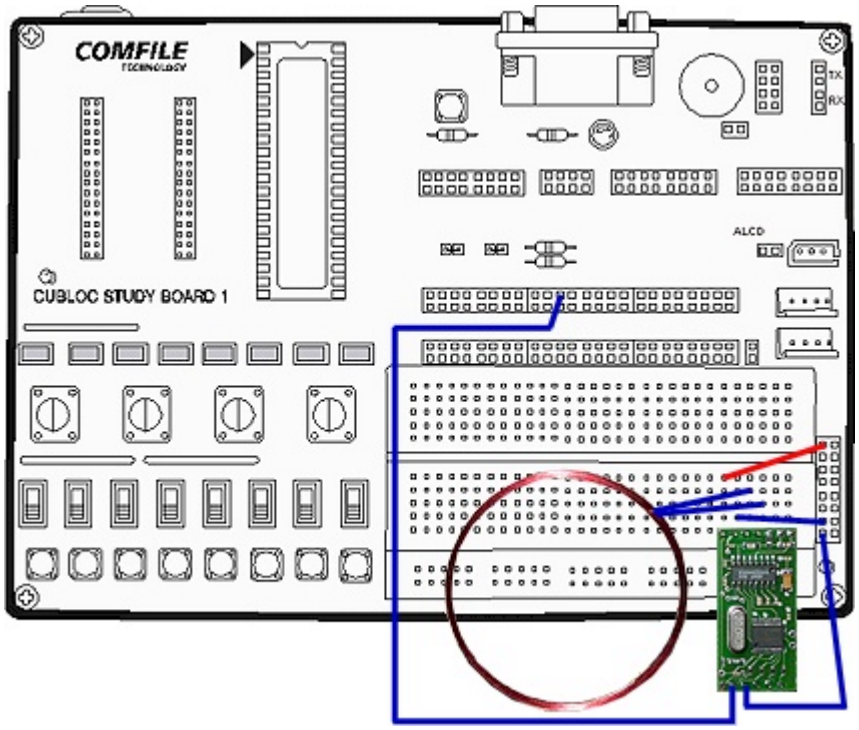
Matériel nécessaire :
- Une platine « CUBLOC Study Board » (facultative)
- 1 module « UM005 » + Antenne RFID + quelques Tag RFID

Préparation matérielle :

Celle-ci repose sur le schéma théorique ci-dessous.



Afin de faciliter la description de cette note d'application, nous utiliserons une platine « CUBLOC Study Board » associée à un module CUBLOC™ CB220 (voir schéma de raccordement ci-après).



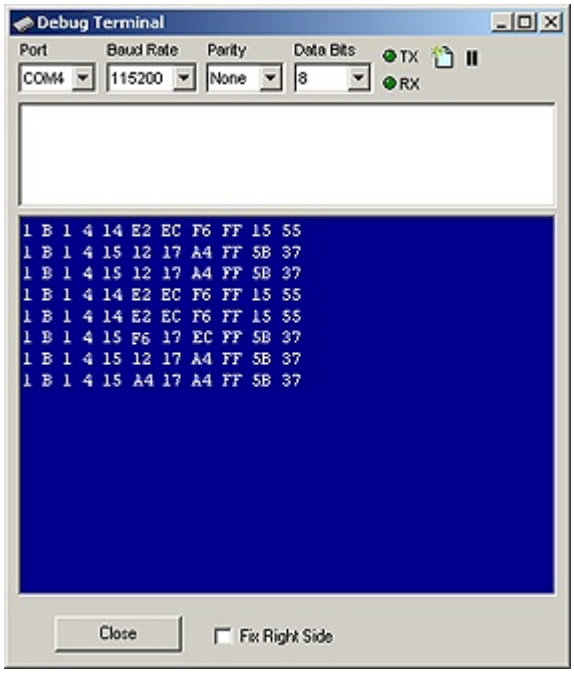
Le module «UM005» sera monté « en l'air » et implanté d'un seul côté sur la plaque de connexion sans soudure. Le raccordement des broches « en l'air » du module pourra se faire au moyen d'un connecteur avec fils au pas de 2.54 mm (réf. BTWF6). On raccordera également l'antenne du module sur la plaque d'essai sans soudure (il existe 2 tailles d'antennes). Dans la pratique, le module « UM005 » devra être câblé au plus près du CB220. De même, l'antenne RFID devra également être câblée au plus près du module « UM005 ».

Cette note d'application comprend 2 programmes : Saisissez en premier lieu celui présenté ci-après (ce dernier est disponible sur notre site Internet : www.lextronic.fr ou sur notre CD-ROM sous le nom « rfid1 »).

Interprétation du premier programme :
Ce dernier vous permettra (à chaque fois que vous passerez un tag près de l'antenne RFID) d'en afficher les 11 octet s de codes pré-mémorisés dans la fenêtre de debug du PC.

Ces octets seront récupérés via une variable de type tableau pour être affichés ensuite un à un dans la fenêtre de debug.

Comme vous pourrez le voir, certains octets de la trame sont identiques alors que d'autres sont différents pour chaque tag.

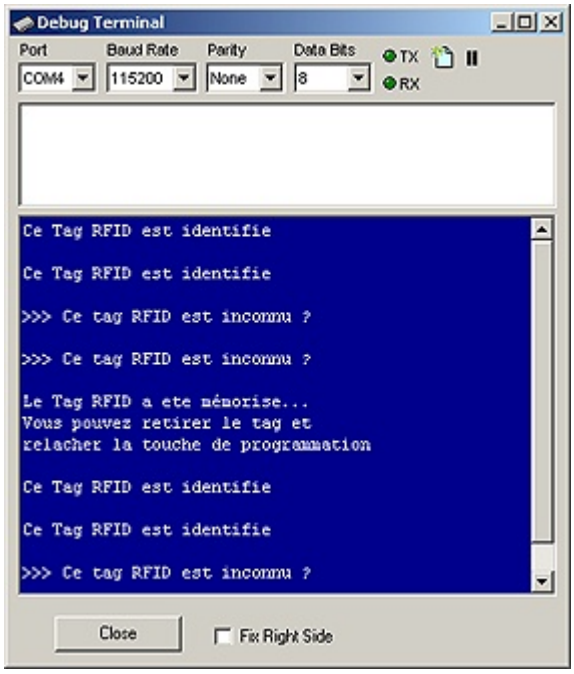



```
'###################################
'#     Gestion d'un module « UM005 »      #
'#     @Lextronic 2006 - 04/03/2006       #
'###################################

Const Device = CB220

Dim code(11) As Byte
Dim a As Byte

Opencom 1,9600,3,30,20
Input 0                                 ' Configure le port P0 en entrée
Delay 1000

Do
   Do While Blen (1,0) = 0              ' Attend la réception du module UM005
   Loop
   Delay 500
   Geta 1,code,11
   For a = 0 To 10                      ' Affiche la suite des codes de la carte
     Debug Hex code(a)," "
   Next
   Debug Cr
 Loop
```

Interprétation du deuxième programme :
Ce dernier est disponible sur notre site Internet : www.lextronic.fr ou sur notre CD-ROM sous le nom « rfid2 »). Il vous permettra d'enregistrer les codes d'un tag RFID dans la mémoire EEPROM du module CUBLOC. Dès lors à chaque fois que vous passerez un tag devant l'antenne du module, l'écran de debug du PC vous indiquera si ce dernier est mémorisé ou si le tag est inconnu.

Pour enregistrer le tag en mémoire, il vous faut au préalable appuyer sur le 1er bouton-poussoir de la platine et passer un tag rfid devant l'antenne. A ce stade, les codes du tag sont récupérés dans une variable de type tableau pour être ensuite stockés en mémoire EEPROM du CUBLOC. Le programme attend ensuite que vous relâchiez le bouton-poussoir (en vous mettant un message associé dans la fenêtre de debug). Passez maintenant les différents tags devant l'antenne RFID. La fenêtre de debug vous indiquera alors si le tag est en mémoire ou non.

Note : Pour chaque tag passé devant l'antenne RFID, les 2 derniers octets retournés par le module « UM005 » correspondent à des octets de contrôle (checksum). Cette note d'application n'effectue pas la vérification de la validité de la trame avec ces octets.



```
'###################################
'#     Gestion d'un module « UM005 »       #
'#     @Lextronic 2006 - 04/03/2006        #
'###################################

Const Device = CB220

Dim code(11) As Byte
Dim a As Byte
Dim acces As Byte

Opencom 1,9600,3,30,20
Input 0                                          ' Configure le port P0 en entrée

Delay 1000

Do
   Do While Blen (1,0) = 0                       ' Attend la réception du module UM005
     Loop
    acces = 0                                    ' initialise variable carte ok
    Delay 500
    Geta 1,code,11                               ' Récupération des codes dans le buffer série
    If Keyinh(0,30) = 1 Then                     ' Test si la touche de programmation est sollicitée
      For a = 0 To 10
       Eewrite a,code(a),1                       ' Mémorisation du code de la carte en mémoire EEprom
      Next
      Debug "Le Tag RFID a ete mémorise...",Cr
      Debug "Vous pouvez retirer le tag et",Cr
      Debug "relacher la touche de programmation",Cr,Cr
      Do While Keyinh(0,30) = 1                  ' Attend relachement de la touche
      Loop
      Delay 1000
      Bclr 1,0                                   ' Efface contenu du buffer de réception

    Else
     For a = 0 To 10                             ' Récupere codes carte et les compare avec codes en
EEPROM
       If code(a) <> Eeread (a,1) Then acces = 1
     Next
     If acces = 0 Then
       Debug "Ce Tag RFID est identifie",Cr,Cr
     Else
       Debug ">>> Ce tag RFID est inconnu ?",Cr,Cr
     End If
     Delay  1000
     Bclr 1,0                                    ' Efface contenu du buffer de réception
    End If
  Loop
```

NOTE D'APPLICATION # 15.
Gestion module pour servomoteurs « SSC03A »

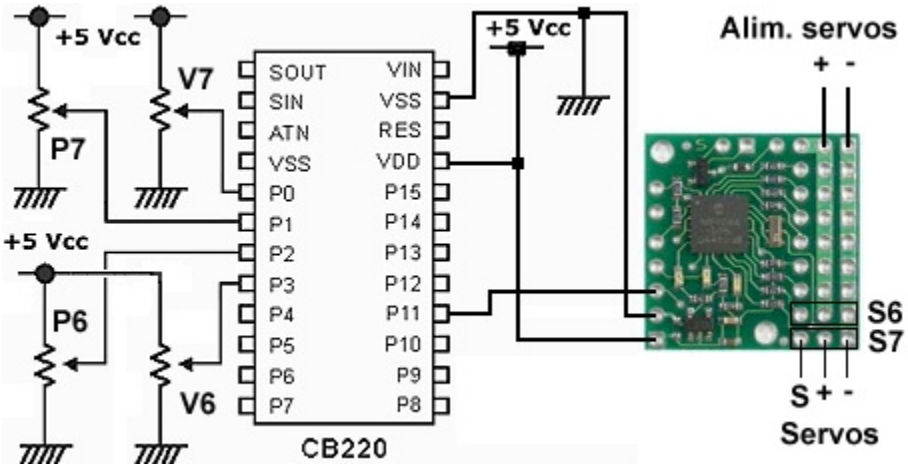
Cette note d'application va vous permettre de piloter un module de gestion de servomoteur « SSC03A » à l'aide d'un CB220. Bien que pouvant directement piloter 3 servomoteurs à l'aide d'un CB220, il peut être utile d'utiliser un petit module additionnel optionnel externe tel que le « SSC03A » afin d'augmenter d'une part le nombre de servomoteurs max. pouvant être pilotés (jusqu'à 128 avec plusieurs platines « SSC03A » et d'autre part afin de pouvoir conserver les ports « PWM » du CB220 pour piloter par exemple des moteurs « cc » (idéal pour la réalisation de robots ludiques).

Le module « SSC03A » s'interface au moyen d'une liaison série. Il dispose de différentes possibilités d'utilisation dont à la faculté de pouvoir gérer et déterminer la position ainsi que la vitesse de déplacement de 1 à 7 servomoteurs.

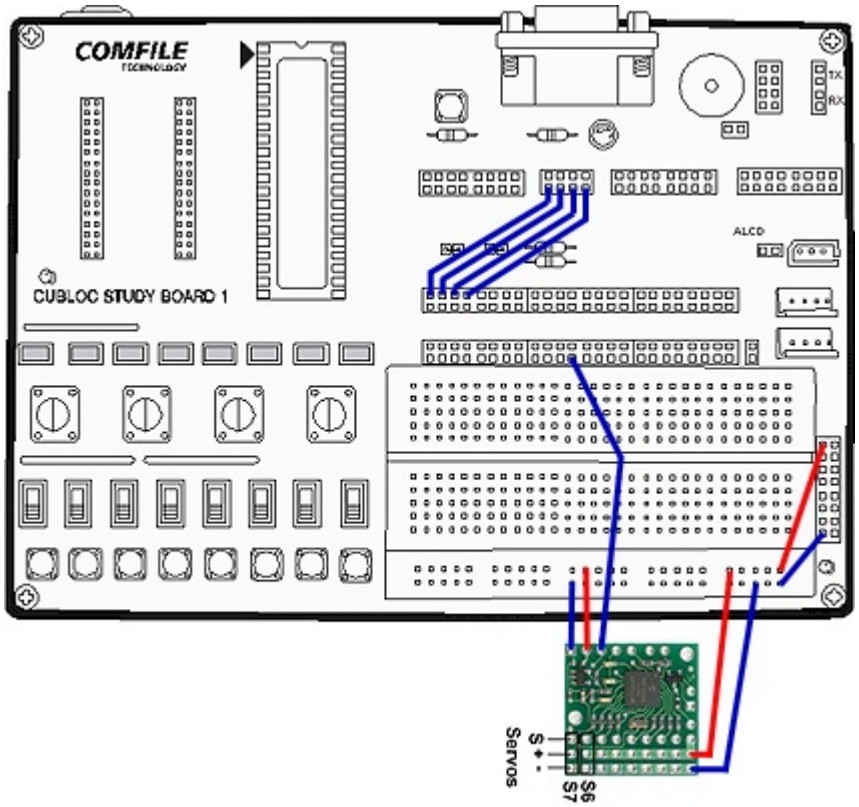
Notions abordées :
- Gestion d'une liaison série.

Matériel nécessaire :
- Une platine « CUBLOC Study Board » (facultative)
- 4 potentiomètres
- 1 à 8 servomoteurs (type modélisme)

Préparation matérielle :
Celle-ci repose sur le schéma théorique ci-dessous.



Afin de faciliter la description de cette note d'application, nous utiliserons une platine « CUBLOC Study Board » associée à un module CUBLOC™ CB220 (voir schéma de raccordement ci-après).



Le module «SSCI» sera monté « en l'air » . Le raccordement des broches de ce dernier pourra se faire au moyen d'un connecteur avec 2 fils au pas de 2.54 mm (réf. BTWF2) et d'un connecteur 3 fils au pas de 2.54 mm (réf. BTWF3). Dans le cadre de notre application, nous n'utiliserons que 2 servomoteurs (petits modèles) ainsi qu'une alimentation commune (celle de la platine) pour le module de pilotage et les servos. Si comme nous vous utilisez la même platine et la même source d'alimentation commune aux servos et la platine, ne connectez pas plus de 2 servomoteurs (de petite puissance) sur le système.

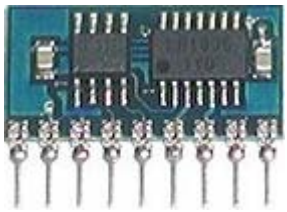
Saisissez ensuite le programme présenté ci-après (ce dernier est disponible sur notre site Internet : www.lextronic.fr ou sur notre CD-ROM sous le nom « ssc03a »).

La petite application vous permettra de modifier indépendamment la position (0 à 180°) de 2 servomoteurs à l'aide des potentiomètres 1 et 3 de la platine. Les servomoteurs sont reliés sur les sorties 6 et 7 du module de commande « SSC03A ». Les potentiomètres 2 et 4 permettront quant à eux de modifier indépendamment la vitesse de déplacement des 2 servomoteurs. Il vous sera ainsi très facilement possible de vérifier que les mouvements des 2 servos sont totalement indépendants (en position et en vitesse).

Pour ce faire, on utilisera le mode de pilotage « Pololu » présent sur le « SSC03A » avec une communication 8 bits (pour avoir accès à ce dernier, il vous faut retirer le cavalier du module « SSC03A »). Le programme débute par l'acquisition des valeurs des 4 potentiomètres, puis par l'envoi au module « SSC03A » de la vitesse de déplacement et de la nouvelle position que devrons prendre les servos. Lors de l'utilisation (afin d'éviter les surchauffes), ne laissez pas les servomoteurs en butée sur les positions extrêmes (il sera ainsi préférable d'ajouter 2 lignes de programmes avant l'envoi des données à la platine « SSC03A », lesquelles pourront tester les valeurs min. et max. afin de reconfigurer la position des servomoteurs en cas de dépassement (par exemple : If pos_serv7 > 230 then pos_serv7 = 230 - > Ainsi vous pourrez plafonner le déplacement).

```
'###################################
'#     Gestion d'un module « SSC03A »    #
'#     @Lextronic 2006 - 04/03/2006        #
'###################################

Const Device = CB220

Dim pos_serv7 As Integer
Dim vit_mot7 As Integer
Dim pos_serv6 As Integer
Dim vit_mot6 As Integer
Dim a As Byte

Opencom 1,9600,3,5,50                          ' Configure le port de communication

'******************** RAZ port  ******************
Input 0                                        ' Configure les ports en entrée
Input 1
Input 2
Input 3

'****************** Boucle principale *************

Do
  pos_serv7 = Adin(0)/4                        ' Récupère valeur 0 - 255 du potentiomètre N° 1
  vit_mot7 = Adin(1)/8                         ' Récupère valeur 0 - 127 du potentiomètre N° 2
  If vit_mot7 = 0 Then vit_mot7 = 1            ' La valeur ne peut pa prendre la valeur 0

  pos_serv6 = Adin(2)/4                        ' Récupère valeur 0 - 255 du potentiomètre N° 3
  vit_mot6 = Adin(3)/8                         ' Récupère valeur 0 - 127 du potentiomètre N° 4
  If vit_mot6 = 0 Then vit_mot6 = 1            ' La valeur ne peut pa prendre la valeur 0

  Putstr 1,&H80,1,1,7,vit_mot7                 ' Envoi l'ordre de vitesse du servo 7
  Delay 10
  a =0
  If pos_serv7 > 128 Then a = 1                ' Configure valeur de data1 (voir notice  SSC03A)
  pos_serv7.BIT7=0                             ' Force le bit 7 de data2 à 0
  Putstr 1,&H80,1,3,7,a,pos_serv7               ' Envoi l'ordre de commande du servomoteur 7
  Delay 10

  Putstr 1,&H80,1,1,6,vit_mot6                 ' Envoi l'ordre de vitesse du servo 6
  Delay 10
  a =0
  If pos_serv6 > 128 Then a = 1                ' Configure valeur de data1 (voir notice  SSC03A)
  pos_serv6.BIT7=0                             ' Force le bit 7 de data2 à 0
  Putstr 1,&H80,1,3,6,a,pos_serv6              ' Envoi l'ordre de commande du servomoteur 6
  Delay 10
Loop
```

NOTE D'APPLICATION # 16.
Gestion des modules « MCM1 / MCM2 »

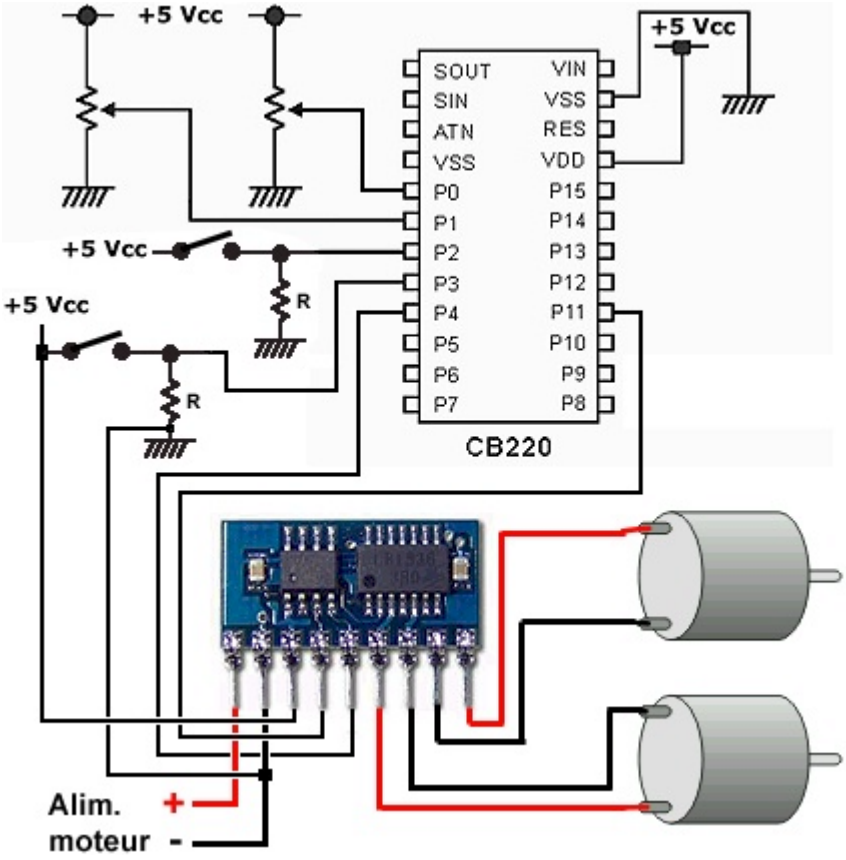


Cette note d'application va vous permettre de piloter des modules de commande de moteurs « cc ». Le premier module utilisé est un « MCM1 ». Ce dernier est capable de gérer 2 moteurs indépendants avec sélection de leur sens de rotation et de leur vitesse. Il s'interface très facilement au moyen d'une liaison série. Bien que le CB220 dispose déjà de 3 sorties PWM, il peut être intéressant d'avoir recours au « MCM1 » pour libérer les sorties PWM afin de pouvoir les exploiter pour le pilotage de servomoteurs par exemple.

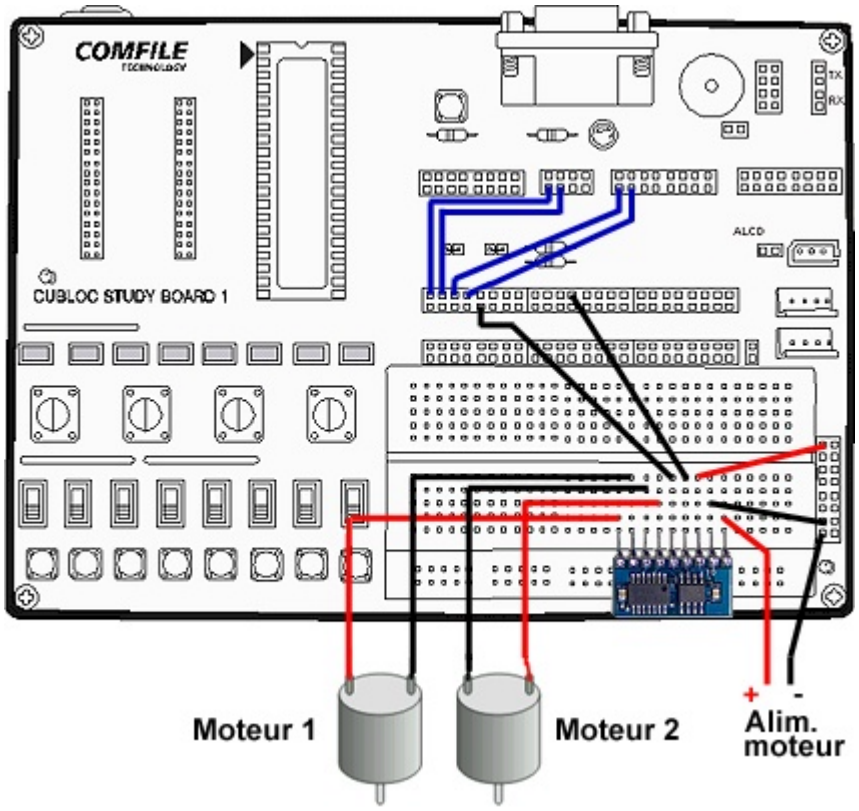
Notions abordées :
- Gestion communication série

Matériel nécessaire :
- Une platine « CUBLOC Study Board » (facultative)
- 1 module « MCM1 » + 2 moteurs « CC » (1,8 à 9 V) + 1 alimentation externe

Préparation matérielle :
Celle-ci repose sur le schéma théorique ci-dessous.



Afin de faciliter la description de cette note d'application, nous utiliserons une platine « CUBLOC Study Board » associée à un module CUBLOC™ CB220 (voir schéma de raccordement ci-après).



La consommation de chaque moteur devra être inférieure à 800 mA env. Utilisez impérativement une alimentation séparée pour les moteurs (si vous réalisez le montage sur la platine « CUBLOC Study Board », ne réutilisez pas l'alimentation du CUBLOC pour alimenter vos moteurs – Réalisez également des connexions les plus courtes possibles). Anti-parasitez les moteurs au moyen de condensateurs de 0,1 uF si nécessaire.

Saisissez ensuite le programme présenté ci-après (ce dernier est disponible sur notre site Internet : www.lextronic.fr ou sur notre CD-ROM sous le nom « mcm1 »).

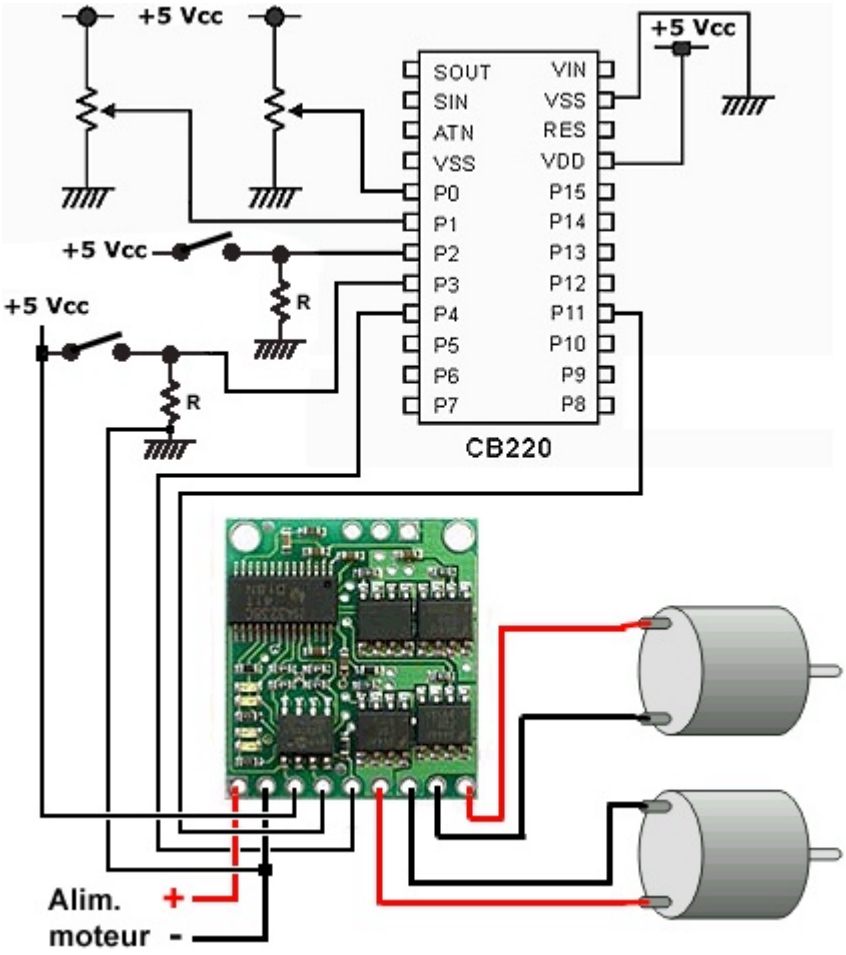
La petite application vous permettra de modifier indépendamment la vitesse de rotation de 2 moteurs à l'aide des 2 premiers potentiomètres de la platine tandis qu'il vous sera également possible de modifier leur sens de rotation à l'aide des 2 premiers interrupteurs. La valeur correspondant à l'image de la vitesse devant être envoyée au « MCM1 » devra être comprise entre 0 et 127 (c'est la raison pour laquelle on divise la valeur de la mesure analogique (normalement comprise entre 0 et 1023) par 8 pour ramener cette valeur entre 0 et 127).

Le module « MCM1 » est interfacé avec 2 fils. Le premier sera relié à la broche TX du CUBLOC (afin de lui envoyer des ordres série) et le second correspond à une entrée de RESET gérée par le module CB220 (un RESET est appliqué en début de programme).

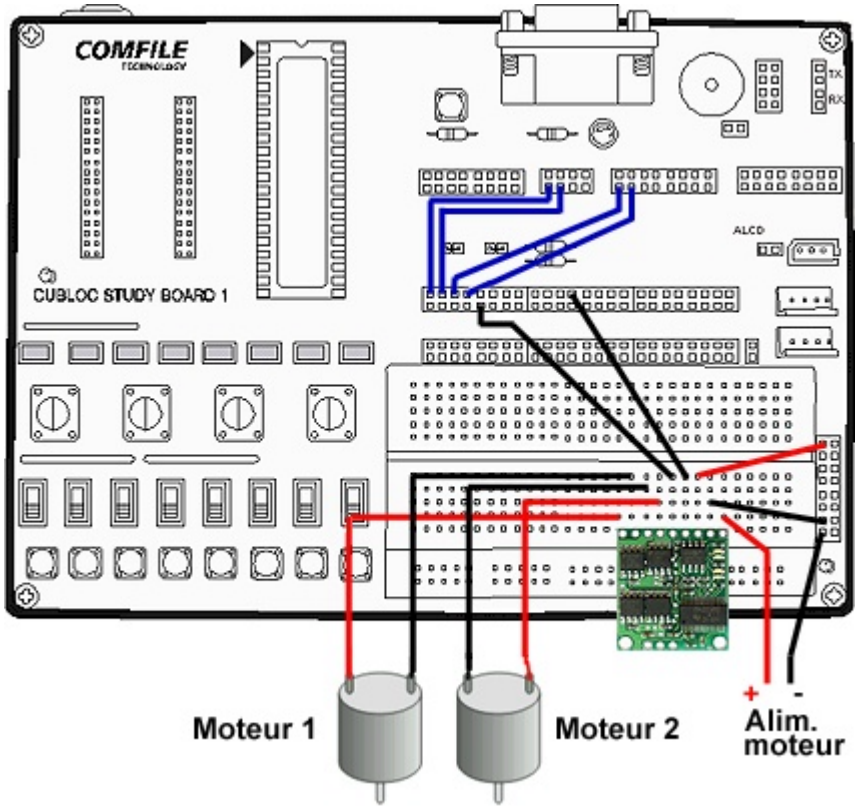
```
'###################################
'#     Gestion d'un module « MCM1 »      #
'#     @Lextronic 2006 - 07/03/2006      #
'###################################

Const Device = CB220

Dim vit_mot1 As Integer
Dim vit_mot2 As Integer
Dim sen_mot1 As Byte
Dim sen_mot2 As Byte

Opencom 1,9600,3,5,50                          ' Configure le port de communication

'******************* RAZ port  *****************
Input 0                                        ' Configure les ports en entrée
Input 1
Input 2
Input 3
High  4                                        ' Configure le port en sortie
Delay 800
Low 4                                          ' Effectue le Reset du module mcm1
Delay 400
High  4

'***************** Boucle principale ************

Do
   vit_mot1 = Adin(0)/8                        ' Récupère valeur 0 - 127 du potentiomètre N° 1
   vit_mot2 = Adin(1)/8                        ' Récupère valeur 0 - 127 du potentiomètre N° 2
   sen_mot1 = In(2)                            ' Récupère position interrupteur 1
   sen_mot2 = In(3)                            ' Récupère position interrupteur 2

   Putstr 1,&H80,0,sen_mot1,vit_mot1           ' Envoi l'ordre de commande du moteur 1
   Delay 100
   Putstr 1,&H80,0,2+sen_mot2,vit_mot2         ' Envoi l'ordre de commande du moteur 2
   Delay 100
Loop
```

A noter qu'il est également possible d'utiliser le module « MCM3 » En vue et place du module « MCM1 ». Ce dernier dispose d'une Compatibilité broche à broche (même si le module n'a pas la même forme). Ce dernier est à même de piloter 2 moteurs 1,6 à 6 Vcc avec une consommation max. en pointe de 4 A (bien qu'il soit conseillé de travailler bien en desssous de cette consommation extreme). Vous pourrez utiliser le même programme que ci-dessus et vous inspirer du schéma théorique et du montage donné ci-dessous.

Utilisez impérativement une alimentation séparée pour les moteurs (si vous réalisez le montage sur la platine « CUBLOC Study Board », ne réutilisez pas l'alimentation du CUBLOC pour alimenter vos moteurs – Réalisez également des connexions les plus courtes possibles. Antiparasitez les moteurs au moyen de condensateurs de 0,1 uF si nécessaire.



Afin de faciliter la description de cette note d'application, nous utiliserons une platine « CUBLOC Study Board » associée à un module CUBLOC™ CB220 (voir schéma de raccordement ci-après). Les fils connexions doivent être les plus courts possibles.



NOTE D'APPLICATION # 17.
Gestion d'un module « ezMOUSE »

Cette note d'application va vous permettre de piloter un module « ezMOUSE ». Ce dernier vous permettra de récupérer les informations d'une souris compatible PS/2 sous la forme de données séries. Vous pourrez ainsi connaître l'état de ses boutons, initialiser sa position, connaître sa position par rapport à la dernière acquisition ou encore sélectionner 4 résolutions. La note d'application consistera à faire afficher un curseur sur un écran LCD de 4 x 20 caractères (raccordé sur le port CuNET) et de gérer le déplacement de ce dernier via la souris.

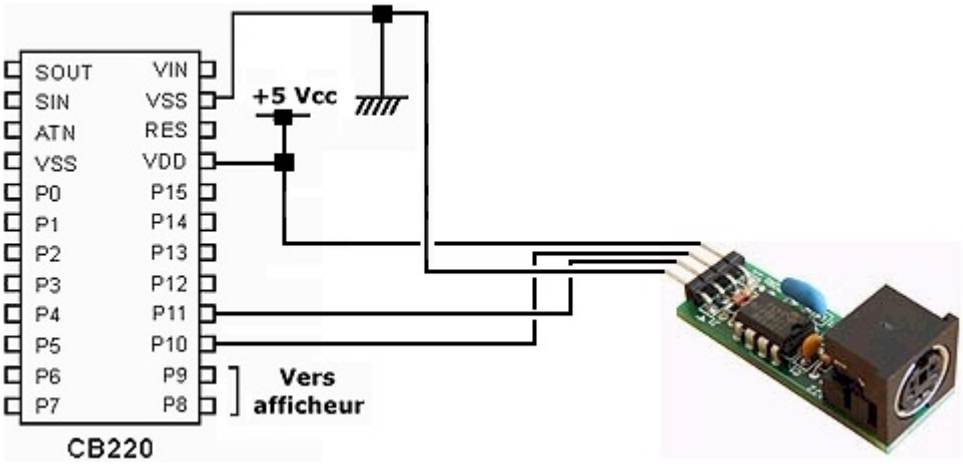
En absence de sollicitation des boutons de la souris, le curseur de l'écran LCD est un « pavé » plein. Ce dernier se transforme en chiffre « 1 », « 2 » ou « 3 » suivant le n° du bouton de la souris que vous solliciterez. De plus si vous essayez de sortir des limites de l'écran, le curseur restera dans les positions extrêmes.

Notions abordées :
- Gestion communication série et I2C™

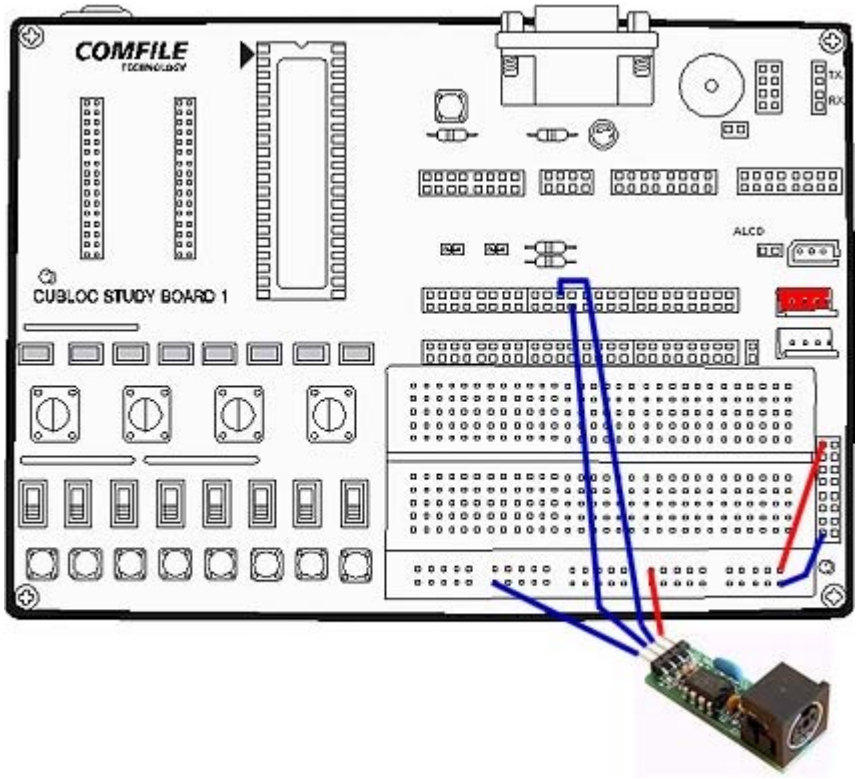
Matériel nécessaire :
- Une platine « CUBLOC Study Board » (facultative)
- 1 module « ezMOUSE »
- 1 Souris compatible PS/2

Préparation matérielle :

Celle-ci repose sur le schéma théorique ci-dessous.



Afin de faciliter la description de cette note d'application, nous utiliserons une platine « CUBLOC Study Board » associée à un module CUBLOC™ CB220 (voir schéma de raccordement ci-après). Ne rallongez pas le fil de la souris. Lors de l'application finale, le module « ezMOUSE » devra être câblé au plus court du module CUBLOC.



Saisissez ensuite le programme présenté ci-après (ce dernier est disponible sur notre site Internet : www.lextronic.fr ou sur notre CD-ROM sous le nom « ezmouse »).

Interprétation du programme :
Le programme se sert de 2 variables pour les coordonnées du curseur (x_lcd et y_lcd) et de 2 variables pour les informations liées aux déplacements de la souris (x_mouse et y_mouse). Le CUBLOC envoi ensuite une demande de données au module « ezMOUSE » et récupère celles-ci (3 octets) dans son buffer de réception série, afin de les analyser. A ce titre, il modifiera la nature du caractère à afficher sur l'écran LCD en fonction des bits 0, 1 et 2 de la variable « bouton » (laquelle mémorise ces informations). La valeur de la variable « caractere » sera alors 49, 50, 51 ou 255 (correspondant respectivement aux caractères « 1 », « 2 », « 3 » - pour montrer quelle touche a été appuyée ou à un carre plein – si aucune touche n'est sollicitée).

Le programme continu en testant si la souris a été bougée. Si c'est le cas, on commence par effacer le caractère présent sur la position actuelle du LCD. On va ensuite tester s'il s'agit d'un déplacement horizontal vers la gauche (dans ce cas, le bit 4 de la variable « bouton » est à 1). Dès lors on diminue la valeur de la positon horizontale du curseur avec la valeur du déplacement horizontal de la souris.

Si la valeur est supérieur à 19, on considère en fait que la valeur du déplacement de la souris a généré un depassement du résultat (en qu'en fait la valeur devrait être négative). Si par exemple la variable correspondant à la position horizontale du curseur sur le LCD « x_lcd » est à 5 (pour rappel, cette variable de type byte ne peut être comprise qu'entre 0 et 255) et que la valeur de déplacement horizontal vers la gauche de la souris est de 7, le résultat de « x_lcd » - 7 sera alors de 254 (les variables de type byte ne gèrent pas les nombres négatifs et suite à une soustraction, en dessous de 0, la variable prend la valeur 255).

Donc par exemple : 5 – 5 = 0 5 – 6 = 255 5 – 7 = 254

C'est pourquoi après la soustraction on effectue le test de la variable par rapport à la valeur 19 (19 correspond à la valeur max. que peu prendre « x_lcd » pour rester sur l'écran). En cas de dépassement, on initialise la valeur de « x_lcd » afin que le curseur reste complètement sur la gauche de l'écran.

La suite du programme effectue le même genre de tests pour le déplacement horizontal droit, vertical haut et bas (avec toujours une limitation de la position du curseur du LCD aux dimensions max. d'affichage de 4 x 20 caractères). La fin de la boucle principale du programme se termine par l'affichage du caractère sur la nouvelle position du LCD.

L'exemple décrit a été réalisé avec une souris « à boule » standard. A noter qu'en fonction du type de souris que vous utiliserez, vous serez peut être amener à modifier la résolution de la mesure de la souris (un ordre série très simple est alors nécessaire – voir la notice du module ezMOUSE). Ceci est surtout le cas avec les souris optiques qui génèrent un déplacement très rapide du curseur en raison de leur résolution améliorée.

```
'#################################
'#   Gestion d'un module « ezMOUSE »  #
'#     @Lextronic 2006 - 08/03/2006       #
'#################################

Const Device = CB220

Dim bouton As Byte
Dim x_mouse As Byte
Dim y_mouse As Byte
Dim x_lcd As Byte
Dim y_lcd As Byte
Dim caractere As Byte

Opencom 1,9600,3,30,20
Set Display 2,0,1,50
Delay 300
Cls
Delay 200
Csroff
Delay 100

x_lcd = 0                                   ' initialise position curseur LCD
y_lcd = 0

Do
  Putstr 1,"G"                              ' Récupère dernière position de la souris
  Delay 20
  bouton = Get(1,1)
```

```
  x_mouse = Get(1,1)
  y_mouse = Get(1,1)

  caractere = 255                          ' Initialise le caractère avec un carre
  If bouton.BIT0 = 1 Then caractere = 49   ' Test si BP N° 1 de la souris sollicité
  If bouton.BIT1 = 1 Then caractere = 51   ' Test si BP N° 3 de la souris sollicité
  If bouton.BIT2 = 1 Then caractere = 50   ' Test si BP N° 2 de la souris sollicité

  If x_mouse <> 0 Or y_mouse <> 0 Then     ' Test si la souris a bougée ?
    Locate x_lcd,y_lcd                     ' Oui -> Efface le caractère de la précédente position
    Print 20                               ' Affiche un espace

   If bouton.BIT4 = 1 Then                 ' Deplacement vers la gauche ?
      x_lcd = x_lcd-x_mouse                ' Modification position horizontale du curseur
      If x_lcd > 19 Then x_lcd = 0         ' test si à fond à gauche ?
   Else                                    ' On est en déplacement vers la droite
      x_lcd = x_lcd+x_mouse                ' Modification position horizontale du curseur
      If x_lcd > 19 Then x_lcd = 19        ' test si à fond à droite ?
   End If

   If bouton.BIT5 = 0 Then                 ' Deplacement vers le haut ?
      y_lcd = y_lcd-y_mouse                ' Modification position verticale du curseur
      If y_lcd > 3 Then y_lcd = 0          ' test si à fond en haut ?
   Else                                    ' On est en déplacement vers le bas
      y_lcd =y_lcd+y_mouse                 ' Modification position verticale du curseur
      If y_lcd > 3 Then y_lcd = 3          ' test si à fond en bas ?
   End If
  End If

  Locate x_lcd,y_lcd                       ' Le curseur prend la nouvelle position
  Print caractere                          ' Affiche le caractère (suivant etat de la souris)
Loop
```

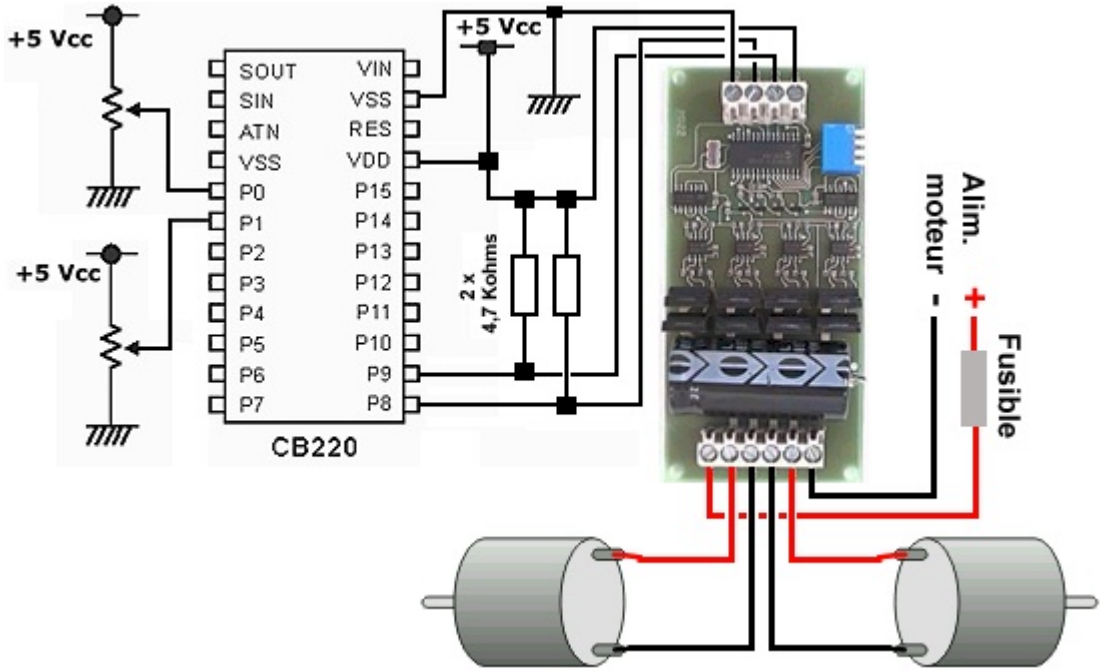
NOTE D'APPLICATION # 18.
Gestion d'un module « MD22 »

Cette note d'application va vous permettre de piloter un module de commande de moteurs 'cc' « MD22 ». Cette carte est un module de puissance avec "pont en H", laquelle dispose de différents modes de commande (via signaux PWM, tension, bus I2C™…). Dans la cadre de cette note d'application, on utilisera Un pilotage via une liaison I2C™ afin de pouvoir régler la vitesse de rotation de 2 moteurs indépendants au moyen de 2 potentiomètres.

Notions abordées :
- Gestion communication série et I2C™

Matériel nécessaire :
- Une platine « CUBLOC Study Board » (facultative)
- 1 module « MD22 »
- 2 moteurs
- 1 alimentation supplémentaire

Préparation matérielle :
Celle-ci repose sur le schéma théorique ci-dessous.



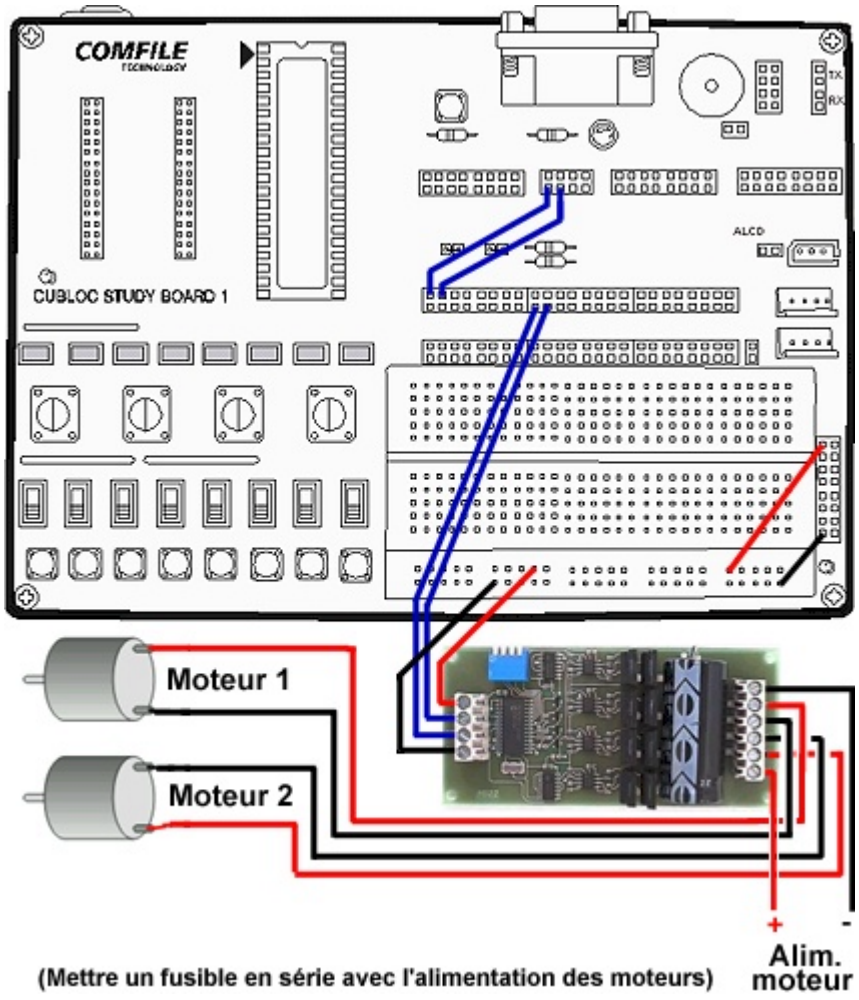
Tous les commutateurs Dils du MD22 doivent être en position « ON » (vers le bas).

Utilisez impérativement une alimentation externe supplémentaire pour les moteurs (en prenant soins d'intercaler impérativement un fusible correctement dimensionné sur l'arrivée positive).

Les moteurs peuvent être alimentés suivant vos modèles sous 5 à 50 Vcc avec une consommation max. en pointe de l'ordre de : 4 à 5 A).

Prévoyez également un anti-parasitage efficace de ces derniers (non représenté ici).

Afin de faciliter la description de cette note d'application, nous utiliserons une platine « CUBLOC Study Board » associée à un module CUBLOC™ CB220 (voir schéma de raccordement ci-après). Utilisez des connexions les plus courtes possibles et respectez les consignes d'alimentation des moteurs énoncés ci-dessus.



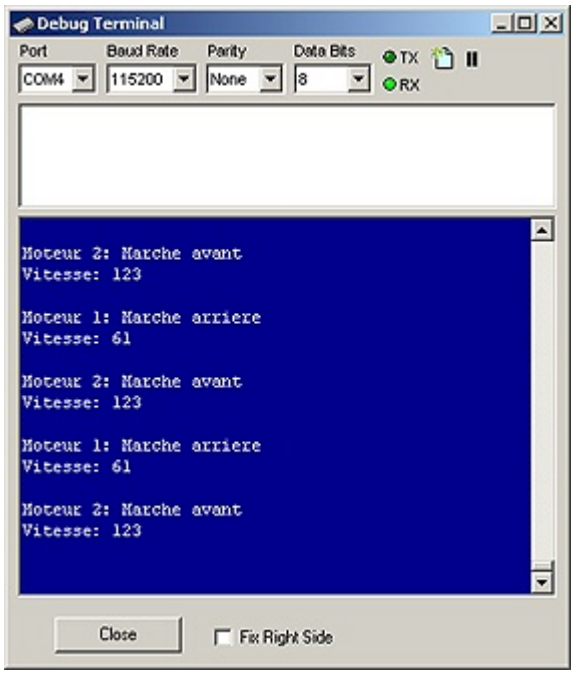
Saisissez ensuite le programme présenté ci-après (ce dernier est disponible sur notre site Internet : www.lextronic.fr ou sur notre CD-ROM sous le nom « md22 »).

Interprétation du programme :
Le programme récupère la valeur de la tension présente sur les curseurs des potentiomètres (la mesure est divisée par 4 afin d'obtenir une valeur comprise entre 0 et 255). Lorsque les potentiomètres sont à mi-course (valeur 128), les moteurs correspondants sont à l'arrêt.

Lorsque vous tournez les potentiomètres vers la droite par rapport à cette position les moteurs corespondant tournent en « marche avant » (valeurs comprises entre 129 et 255).

A l'inverse, lorsque vous tournez les potentiomètres vers la gauche par rapport à la position de mi-course, les moteurs corespondant tournent en « marche arrière » (valeur comprises entre 0 et 127).

La fenêtre DEGUG du PC vous affichera également le sens de rotation de chaque moteur et une valeur représentant une « image » de leur vitesse de rotation (via les variables « vit1 » et « vit2 »).




```
'###################################
'#     Gestion d'un module « MD22 »       #
'#     @Lextronic 2006 - 08/03/2006       #
'###################################

Const Device = CB220
Dim errorcom As Byte
Dim vit1 As Integer
Dim vit2 As Integer

Set I2c 8,9

'****************** RAZ port  *****************
Input 0                                                    ' Configure les ports en entrée
Input 1
```

```
'***************** Boucle principale *************

Do
  vit1 = Adin(0)/4                          ' Récupère valeur 0 - 255 du potentiomètre N° 1
  vit2 = Adin(1)/4                          ' Récupère valeur 0 - 255 du potentiomètre N° 2

  I2cstart                                  ' Condition Start I2C
  errorcom = I2cwrite (&HB0)                ' Adresse du module "MD22"
  errorcom = I2cwrite (&H01)                ' Selectionne l'adresse de la vitesse du moteur 1
  errorcom = I2cwrite (vit1)                ' Envoi la vitesse de rotation
  I2cstop                                   ' Condition Stop I2C

  If vit1 > 128 Then
     vit1.BIT7=0
     Debug "Moteur 1: Marche avant",Cr,"Vitesse: ",Dec vit1,Cr,Cr
  Else
     vit1 = 128 - vit1
     Debug "Moteur 1: Marche arriere",Cr,"Vitesse: ",Dec vit1,Cr,Cr
  End If

   I2cstart                                 ' Condition Start I2C
  errorcom = I2cwrite (&HB0)                ' Adresse du module "MD22"
  errorcom = I2cwrite (&H02)                ' Selectionne l'adresse de la vitesse du moteur 2
  errorcom = I2cwrite (vit2)                ' Envoi la vitesse de rotation
  I2cstop                                   ' Condition Stop I2C

  If vit2 > 128 Then
     vit2.BIT7=0
     Debug "Moteur 2: Marche avant",Cr,"Vitesse: ",Dec vit2,Cr,Cr
  Else
     vit2 = 128 - vit2
     Debug "Moteur 2: Marche arriere",Cr,"Vitesse: ",Dec vit2,Cr,Cr
  End If
Loop
```

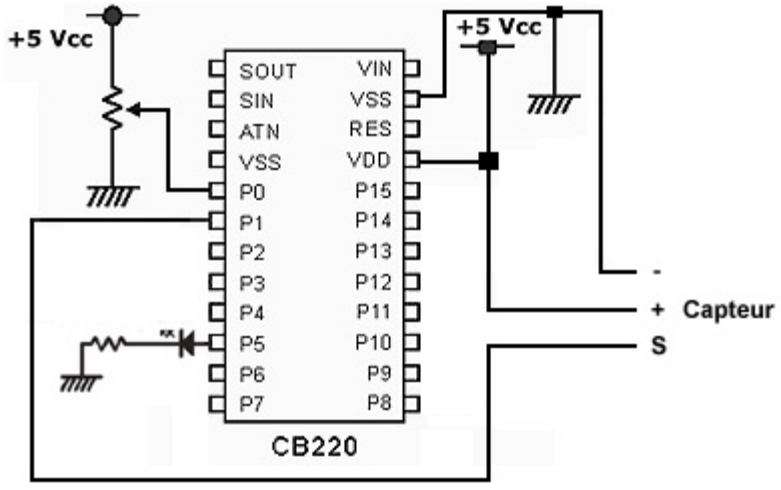
NOTE D'APPLICATION # 19.
Expérimentations avec différents capteurs

Cette note d'application va vous montrer comment interfacer différents types de capteurs avec votre module CUBLOC. Pour ce faire, on utilisera 2 entrées de conversion analogique/numérique. La première sera reliée sur sur le curseur d'un potentiomètre tandis que la seconde sera reliée sur la sortie du capteur. Un afficheur LCD Comfile à commande I2C™ sera utilisé pour afficher une « représentation » de la valeur des tensions présentes sur les 2 entrées de conversion analogique/numérique sous la forme de 2 bargraphs (voir explications sur le fonctionnement du bargraph sur la note d'application N# 1). Le potentiomètre représenté par le bargraph en haut du LCD servira à régler un seuil de déclenchement qui lorsqu'il sera déassé par la tension du capteur, activera l'allumage d'une Led connectée sur un port P5 du CUBLOC configuré en sortie. Pour les besoins de la note d'application on pourra utiliser des capteurs type « télémètre infrarouge » GP2D120 ou des capteurs de la série « Phidgets™ ».

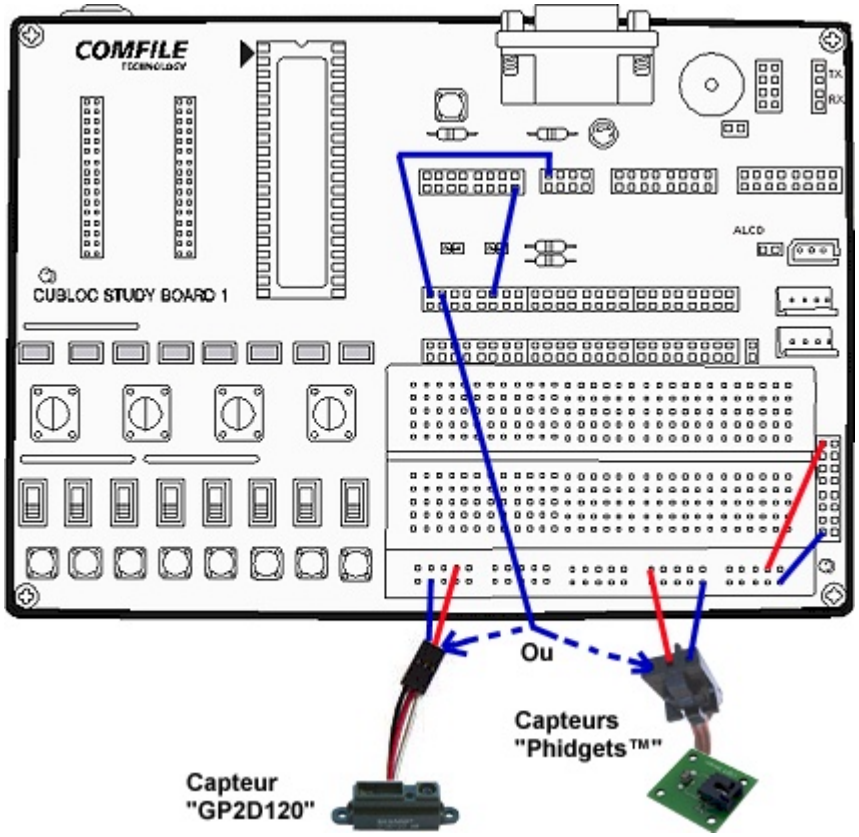
Notions abordées :
- Gestion d'un écran « LCD » à commande I2C™ (avec redéfinition de caractères)
- Gestion d'entrées de conversion « Analogique / Numérique »

Matériel nécessaire :
- Une platine « CUBLOC Study Board » (facultative)
- Différents types de capteurs
- Une Led

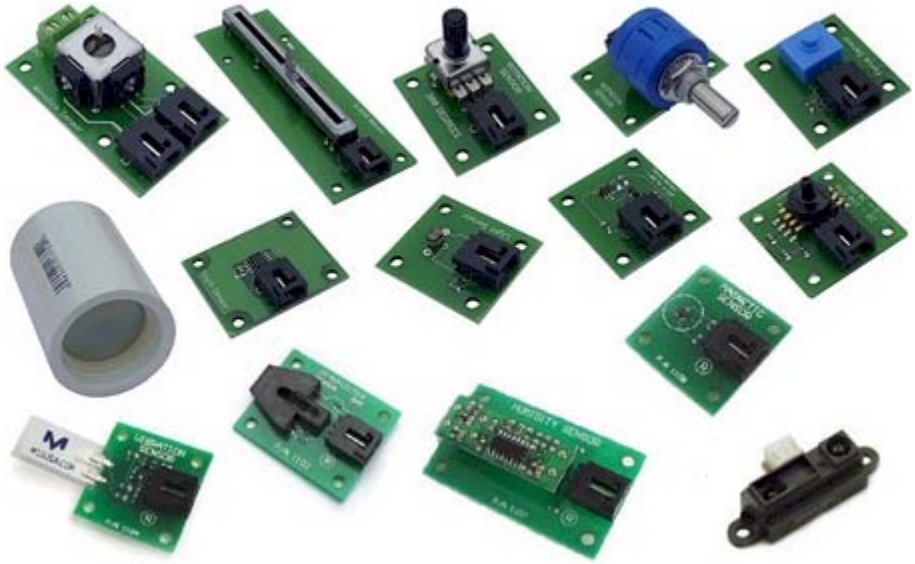
Préparation matérielle :
Celle-ci repose sur le schéma théorique ci-dessous.



Afin de faciliter la description de cette note d'application, nous utiliserons une platine « CUBLOC Study Board » associée à un module CUBLOC™ CB220.



Le schéma donne 2 types de câblage suivant que vous utilisez un télémètre infrarouge de type « GP2D120 » ou un capteur de type « Phidgets ». Les capteurs « Phidgets » se présentent sous la forme d'une petite platine aux fonctions diverses (suivant les modèles) avec un câble de raccordement associé à un petit connecteur.



Il vous sera ainsi possible d'évaluer différents type de technologies en associant votre module CUBLOC à des capteurs de force, de température, de lumière, de champ magnétique, de vibration, d'humidité, de pression, de variation de résistance (via potentiomètres), de proximité (via fourche infrarouge), de détection capacitive, de détection infrarouge passif, etc…

Saisissez ensuite le programme présenté ci-après (ce dernier est disponible sur notre site Internet : www.lextronic.fr ou sur notre CD-ROM sous le nom « ir1 » - pour les capteurs « Phidgets™ » ou « Gp2d1 » pour le capteur « GP2D120 »).

Le programme est très simple à comprendre et n'apporte pas de commentaire particulier. La différence entre les 2 programme réside dans la condition de prise en compt (> ou <) de la tension de référence. On notera aussi que le seuil de déclenchement entre le bargraph du haut et du bas peut présenter un legère différence, due à la résolution et à la différence d'échelle d'affichage des 2 tensions.

```
'###################################
'#    Gestion de capteurs divers          #
'#    @Lextronic 2006 - 11/03/2006        #
'###################################

Const Device = CB220
Set Display 2,0,1,50                                    ' Initialisation de l'afficheur LCD

Dim pot As Integer
Dim consigne As Integer
Delay 100
Cls
Delay 200
Csroff
Delay 100

Print &H1B,&H44,8,0,0,0,0,0,0,0                 ' Redéfinition des caractères du bargraph
Print &H1B,&H44,9,16,16,16,16,16,16,16,16
Print &H1B,&H44,10,24,24,24,24,24,24,24,24
Print &H1B,&H44,11,28,28,28,28,28,28,28,28
Print &H1B,&H44,12,30,30,30,30,30,30,30,30

Input 0                                 ' Configure le port P0 de conversion analogique/numérique en entrée
Input 1                                 ' Configure le port P1 de conversion analogique/numérique en entrée
Low 5                                   ' Configure port P5 en sortie et la force au niveau logique BAS
Do
   pot = Adin(1)
   consigne = pot                       ' Mémorise la valeur de consigne
   bargraph 1                           ' Affiche le bargraph de la valeur de consigne
   pot = Adin(0)
   pot=pot*1.2

   If pot > consigne Then           ' Test si l'obstacle est à une distance inférieur de la valeur de consigne ?
      Out 5,1                       ' Oui -> Allume la Led
    Else
      Out 5,0                       ' Non -> Etteind la Led
   End If
   bargraph 0                       ' Affiche le bargraph du capteur IR
   Loop
End


‘ Recopier ici la sous routine BARGRAPH du programme de la note d'application N# 1
```

NOTE D'APPLICATION # 20.
Pilotage de servomoteurs via sorties « PWM »

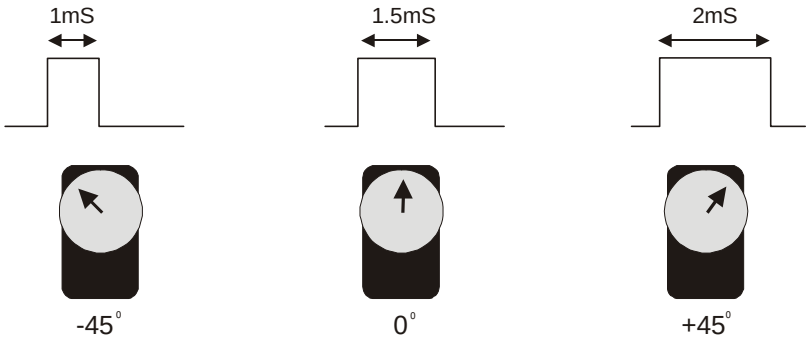
Il est très facile de piloter jusqu'à 3 servomoteurs avec le CUBLOC « CB220 » et jusqu'à 6 servomoteurs avec les modèles « CB280 » et « CB290 » au moyen des sorties « PWM ». Ces servomoteurs utilisés pour le modélisme pourront être idéalement exploité pour la réalisation de robots ludique (afin de déplacer leurs bras ou pattes par exemple). Les servomoteurs utilisent 3 fils. Le fil noir correspond à la masse, le rouge au + (alimentation 4,8 à 6 Vcc) et le fil jaune (ou blanc) au signal de Pilotage proprement dit.

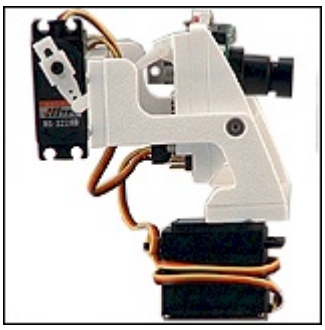
Ce signal « PWM » modifiera la position du palonnier selon la valeur de son rapport cyclique. Le signal en question devra être généré en permanence afin que le servomoteur conserve sa position.

Une impulsion de 1 ms positionnera le palonnier à - 45°.

Une impulsion de 1,5 ms positionnera le palonnier à 0°.

Une impulsion de 2 ms positionnera le palonnier à + 45 °.





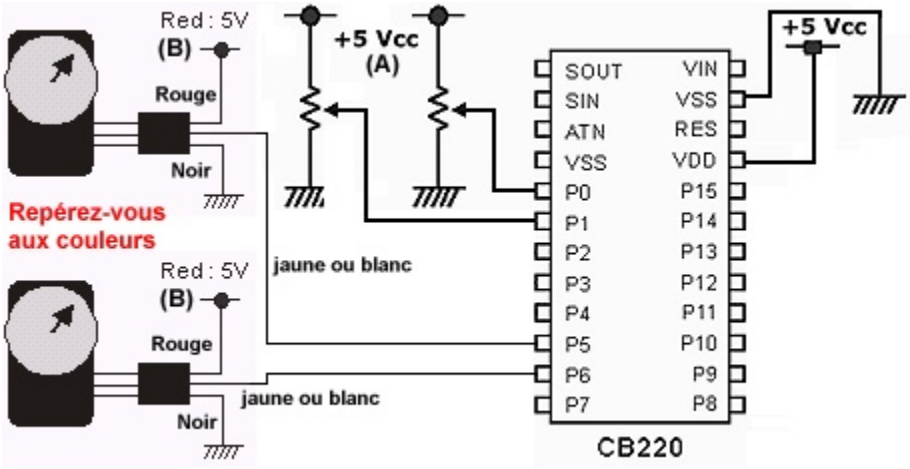
Il est possible d'observer quelques legères variations de position en fonction de la marque des servos que vous utiliserez. La petite note d'application ci-dessous vous permettra de piloter la position de 2 servomoteurs à l'aide de 2 potentiomètres. Vous pourrez ainsi par exemple piloter une tourelle « Pan / Tilt » au moyen de 2 servomoteurs via des potentiomètres ou un « joystick x-y » afin de pouvoir déplacer une caméra de surveillance ou une caméra de reconnaissance de couleur telle que la « CMUcam » par exemple.

Notions abordées :
- Gestion d'un servomoteurs via des signaux « PWM »
- Gestion d'entrées de conversion « Analogique / Numérique »

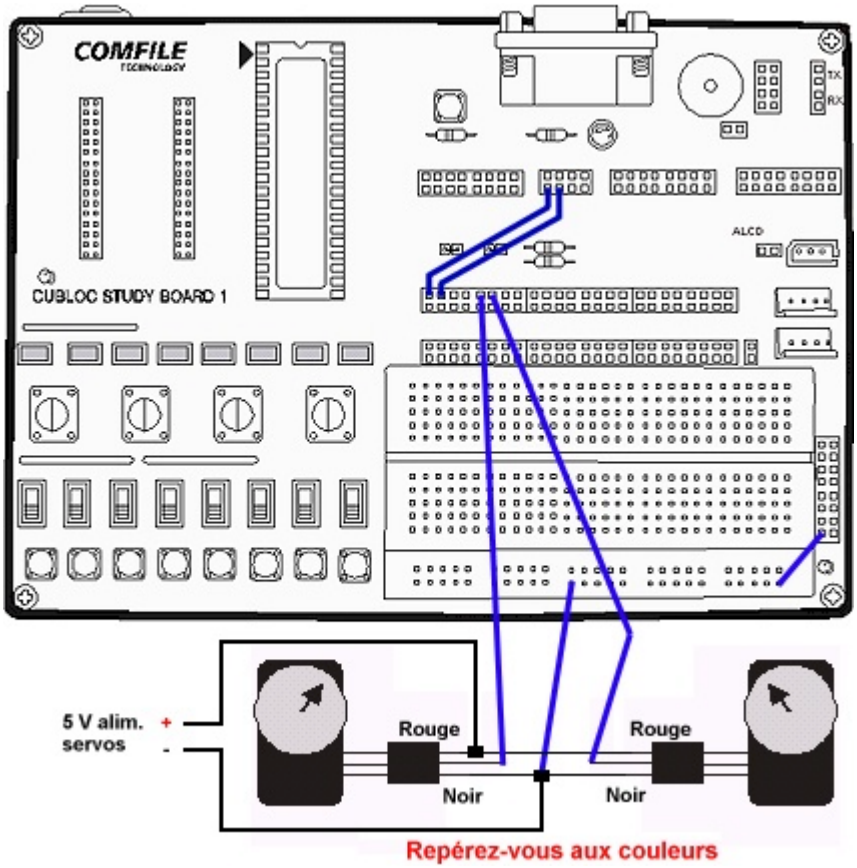
Matériel nécessaire :
- Une platine « CUBLOC Study Board » (facultative) + 2 servomoteurs

Préparation matérielle :

Celle-ci repose sur le schéma théorique ci-dessous.



Afin de faciliter la description de cette note d'application, nous utiliserons une platine « CUBLOC Study Board » associée à un module CUBLOC™ CB220.



Dans tous les cas, suivez la correspondance des couleurs des connecteurs des servomoteurs (et non pas l'ordre des broches du connecteur qui ne reflète pas le « bon brochage »). Si vous utilisez la platine « CUBLOC Study Board », alimentez les servomoteurs via une source externe (n'utilisez pas l'alimentation de la platine sauf si vous ne piloter qu'un petit servomoteur).

Saisissez ensuite le programme présenté ci-après (ce dernier est disponible sur notre site Internet : www.lextronic.fr ou sur notre CD-ROM sous le nom « servo ».

Les servomoteurs sont normalement prévus pour bénéficier d'une rotation de leur palonnier sur 90 °. Pour y parvenir, il vous suffira de générer un signal « PWM » d' 1 ms pour dispose de la position - 45 ° avec la commande **Pwm 0,2500,32768**.

Pour amener le palonnier en position + 45 °, il vous suffira de générer un signal « PWM » de 2 ms avec la commande **Pwm 0,4000,32768**.

Dans la pratique (surtout si vous voulez utiliser les servomoteurs pour réaliser des petits robots ludiques), il est possible d'obtenir une course de rotation de 180 ° du palonnier en utilisant les commandes **Pwm 0,2350,32768** et **Pwm 0,4300,32768**. Toutefois utilisez ces valeurs avec précautions. Comme indiqué précédemment, ces valeurs peuvent varier en fonction de la marque de vos servomoteurs. Vérifiez donc impérativement que les servomoteurs n'arrivent jamais en buté (risque de destruction, de surchauffe et de destruction).

Le petit programme ci-dessous va faire varier alternativement (et de façon opposée) les servomoteurs sur 180 ° (de la position - 90 ° à + 90 °).

```
'###################################
'#     Gestion de servomoteurs             #
'#     @Lextronic 2006 - 11/03/2006        #
'###################################

Const Device = CB220

Input 0                                 ' Configure les ports de conversion analogique/numérique en entrée
Input 1
Low 5                                                   ' Configure les port P5 et P6 en sortie
Low 6

Do
  Pwm 1,4300,32768                      ‘ Servo 1 en position +90 °
  Pwm 0,2350,32768                      ‘ Servo 2 en position - 90 °
  Delay 1000
  Pwm 1,2350,32768                      ‘ Servo 1 en position - 90 °
  Pwm 0,4300,32768                      ‘ Servo 2 en position +90 °
  Delay 1000
Loop
```

Saisissez ensuite le second programme présenté ci-après (ce dernier est disponible sur notre site Internet : www.lextronic.fr ou sur notre CD-ROM sous le nom « servo2 ».

Ce programme vous permet de modifier la position de 2 servomoteurs à l'aide de 2 potentiomètres. Après avoir récupéré l'image de la valeur analogique présente sur le curseur des potentiomètres, on effectue une remise à l'échelle afin que pour une variation de 0 à 1023, on obtienne une variation de 2350 à 4300 (ce qui correspond aux valeurs extrêmes des servomoteurs). Attention, en fonction de la marque de vos servomoteurs, vous pourrez être amené à modifier quelque peu le calcul de mise à l'échelle afin d'éviter que les servomoteurs ne soient en buté.

```
'###################################
'#    Gestion de servomoteurs      #
'#    @Lextronic 2006 - 11/03/2006 #
'###################################

Const Device = CB220

Dim position As Integer
Dim calcul As Single

Input 0                             ' Configure les ports de conversion analogique/numérique en entrée
Input 1
Low 5                               ' Configure les port P5 et P6 en sortie
Low 6

Do
  calcul = Adin(0) * 1.91           ' Remise à l'echelle de la valeur du potentiometre
  position = calcul + 2300          ' Conversion en "integer" + valeur talon servo
  Pwm 0,position,32768

  calcul = Adin(1) * 1.91           ' Remise à l'echelle de la valeur du potentiometre
  position = calcul + 2300          ' Conversion en "integer" + valeur talon servo
  Pwm 1,position,32768
Loop
```

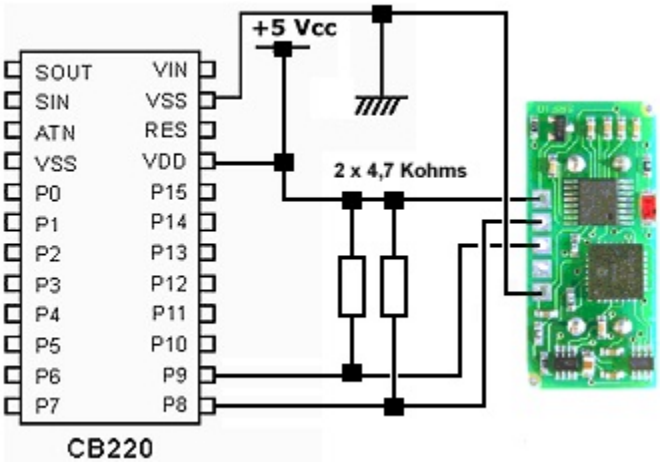
NOTE D'APPLICATION # 21.
Gestion d'un télémètre ultrason « MSU10 »

Cette note d'application va vous permettre de piloter un module télémètre ultrason « MSU10 ». Ce petit module, idéalement conçu pour les applications liées à la robotique ludique, est capable de déterminer la distance qui le sépare d'un obstacle se présentant devant lui (entre 3 cm et 6 m). Doté de 2 cellules ultrason, son principe de fonctionnement repose sur celui des "sonars". Il s'interface à l'aide de son bus I2C™ et se pilote à la manière d'une mémoire EEPROM type 24xx. Ce dernier peut vous retourner la valeur de la distance en "mm", en "inch" ou sous forme d'une durée (en µs) liée à l'echo de l'émetteur ultrason. A noter enfin qu'il vous sera possible d'adresser jusqu'à 16 modules différents par le bus I2C™. A noter que le « MSU08 » n'est destiné qu'à un usage ludique (pour la construction de petits robots mobiles capable d'éviter les obstacles).

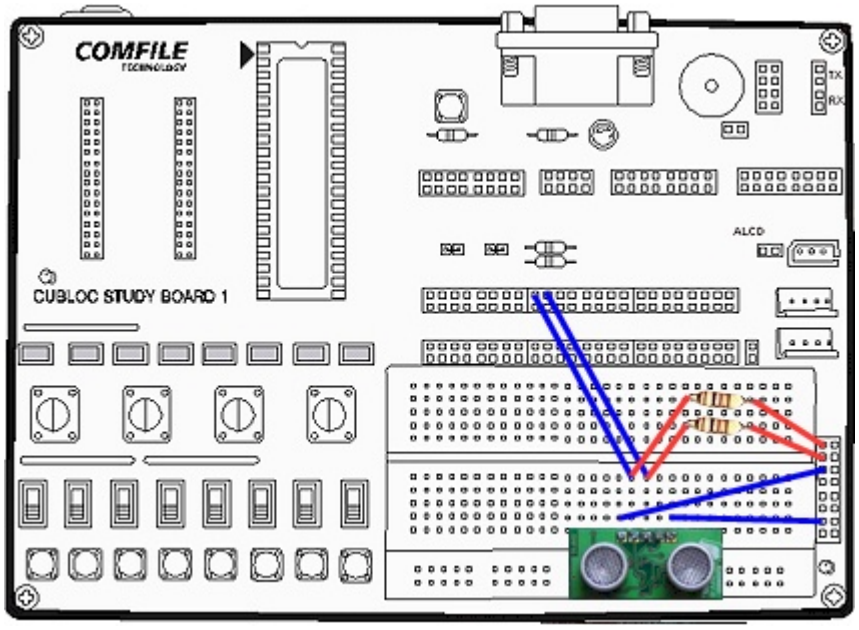
Notions abordées :
- Gestion I2C™

Matériel nécessaire :
- Une platine « CUBLOC Study Board » (facultative)
- 1 module « MSU10 »

Préparation matérielle :
Celle-ci repose sur le schéma théorique ci-dessous.



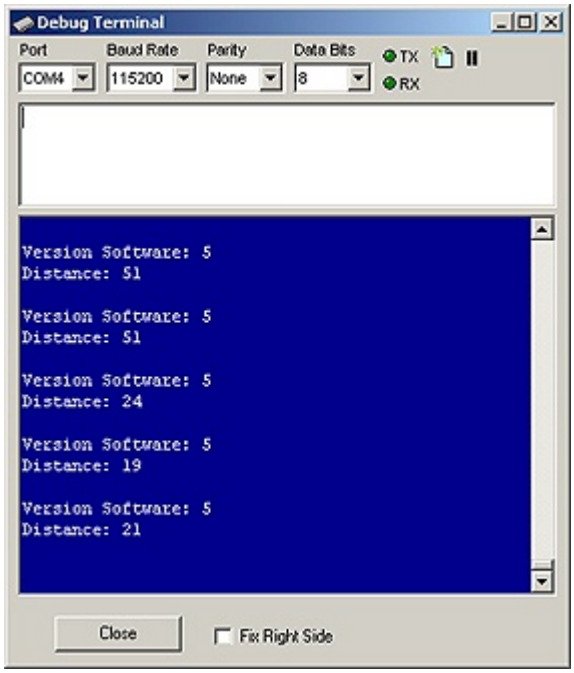
Afin de faciliter la description de cette note d'application, nous utiliserons une platine « CUBLOC Study Board » associée à un module CUBLOC™ CB220 (voir schéma de raccordement ci-après).



Les résistances ont pour valeur 4,7 Kohms.

Saisissez ensuite le petit programme présenté ci-dessus (ce dernier est disponible sur notre site Internet : www.lextronic.fr ou sur notre CD-ROM sous le nom « msu10 »).

Ce programme va permettre de venir lire les registres du module « MSU10 » après avoir initié une demande de mesure en cm. Dès lors, vous pourrez voir apparaître dans la fenêtre de DEBUG le N° du firmware du module « MSU10 » ainsi que la distance en cm qui sépare le module d'un obstacle.



```
'###################################
'#     Gestion de servomoteurs     #
'#     @Lextronic 2006 - 11/03/2006     #
'###################################

Const Device = CB220
Dim errorcom As Byte
Dim version As Byte
Dim range As Integer
Set I2c 8,9

Do
   I2cstart                        ' Condition Start I2C
   errorcom = I2cwrite (&HE0)      ' Adresse du module MSU10
   errorcom = I2cwrite (&H00)      ' Selectionne l'adresse du registre de commande
   errorcom = I2cwrite (&H51)      ' Active mesure en cm
  I2cstop                          ' Condition Stop I2C
  Delay 100                        ' Tempo
  I2cstart                         ' Condition Start I2C
  errorcom = I2cwrite (&HE0)       ' Adresse du module MSU10
  errorcom = I2cwrite (0)          ' Sélectionne l'adresse du premier registre à lire
  I2cstart                         ' Condition Start I2C
  errorcom = I2cwrite (&HE1)       ' Sélectionne condition de lecture I2C
  version = I2cread(0)             ' Récupere N° de révision du module MSU10
  errorcom = I2cwrite (&HE1)       ' Sélectionne condition de lecture I2C
  range = I2cread(0)               ' Lecture adresse 1 (non utilisée)
  errorcom = I2cwrite (&HE1)       ' Sélectionne condition de lecture I2C
  range.byte1=I2cread(0)           ' Récupère octet poids fort de la distance
  errorcom = I2cwrite (&HE1)       ' Sélectionne condition de lecture I2C
  range.byte0=I2cread(0)           ' Récupère octet poids faible de la distance
I2cstop                            ' Condition Stop I2C

Debug "Version Software: ",Dec version,Cr
Debug "Distance: ",Dec range,Cr,Cr
Delay 500
Loop
```

Gestion d'un télémètre ultrason « MSU235 »

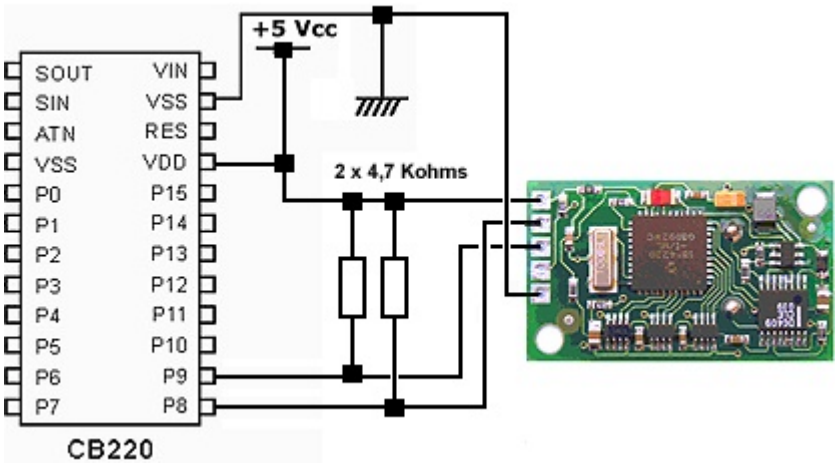
Il est également possible d'utiliser selon le même schéma et avec le même programme le télémètre ultrason « MSU235 ». Cet autre petit module (conçu pour les applications liées à la robotique ludique, est capable de déterminer la distance qui le sépare d'un obstacle se présentant devant lui (entre 10 cm et 1,2 m mais avec un angle restreint de seulement 15°). Doté d'une seule cellule ultrason, son principe de fonctionnement repose sur celui des "sonars". Il s'interface à l'aide de son bus I2C™ et se pilote à la manière d'une mémoire EEPROM

Vous trouverez ci-dessous le schéma type d'utilisation avec un module CB220 ainsi que le schéma d'utilisation sur une platine « CUBLOC Study Board ».

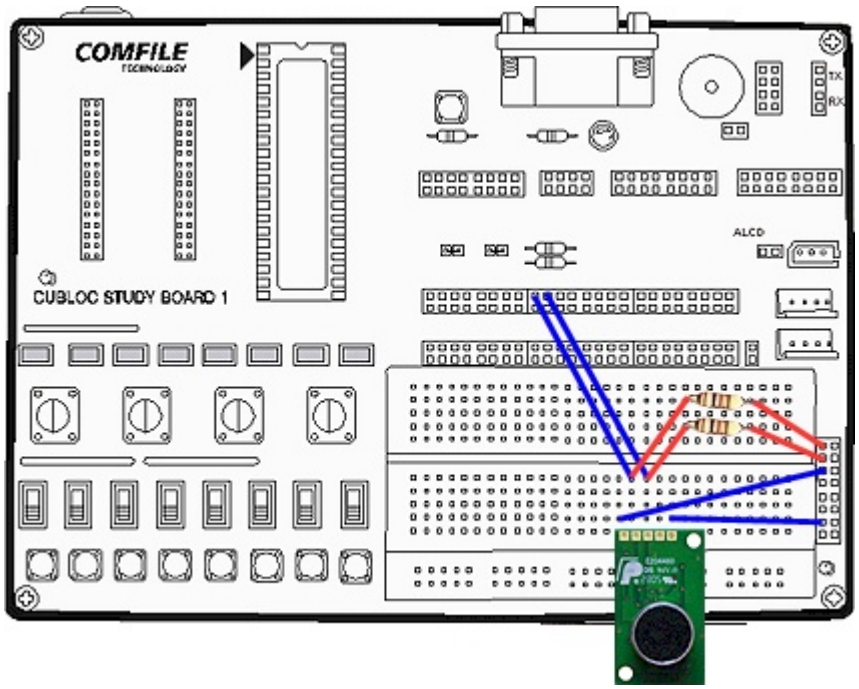
Le programme reste le même que celui du module « MSU10 ».

Préparation matérielle :

Celle-ci repose sur le schéma théorique ci-dessous.



Afin de faciliter la description de cette note d'application, nous utiliserons une platine « CUBLOC Study Board » associée à un module CUBLOC™ CB220 (voir schéma de raccordement ci-après).



NOTE D'APPLICATION # 22.
Gestion module pour servomoteurs « SMCPRO »

Cette note d'application va vous permettre de piloter un module de gestion de servomoteur « SSMCPRO » à l'aide d'un CB220. Bien que pouvant directement piloter 3 servomoteurs à l'aide d'un CB220, il peut être utile d'utiliser un petit module additionnel optionnel externe tel que le « SMCPRO » afin d'augmenter d'une part le nombre de servomoteurs max. pouvant être pilotés (jusqu'à 256 avec plusieurs platines « SMCPRO » et d'autre part afin de pouvoir conserver les ports « PWM » du CB220 pour piloter par exemple des moteurs « cc » (idéal pour la réalisation de robots ludiques).

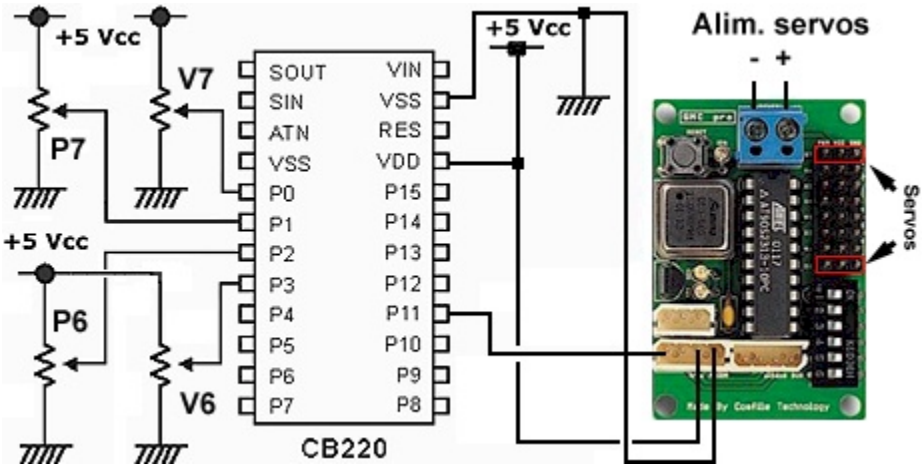
Le module « SMCPRO » s'interface au moyen d'une liaison série. Il dispose de différentes possibilités d'utilisation dont à la faculté de pouvoir gérer et déterminer la position ainsi que la vitesse de déplacement de 1 à 7 servomoteurs.

Notions abordées :
- Gestion d'une liaison série.

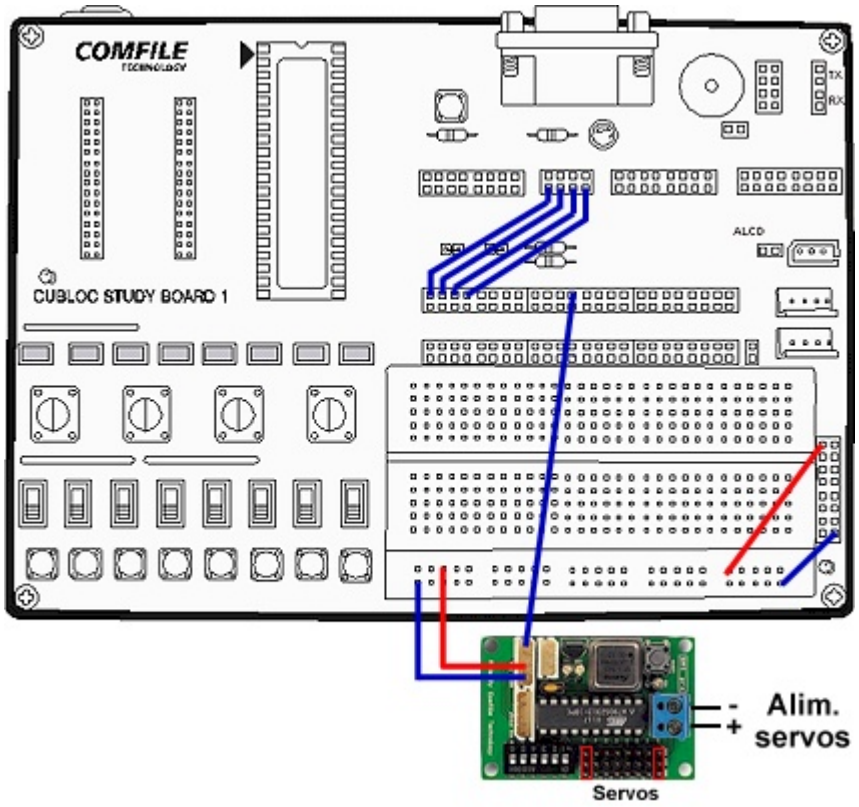
Matériel nécessaire :
- Une platine « CUBLOC Study Board » (facultative)
- 4 potentiomètres
- 1 à 8 servomoteurs (type modélisme)

Préparation matérielle :

Celle-ci repose sur le schéma théorique ci-dessous.



Afin de faciliter la description de cette note d'application, nous utiliserons une platine « CUBLOC Study Board » associée à un module CUBLOC™ CB220 (voir schéma de raccordement ci-après).



Dans le cadre de notre application, nous n'utiliserons que 2 servomoteurs (petits modèles) ainsi qu'une alimentation externe (il est toutefois possible d'utiliser l'alimentation de la platine « CUBLOC Study Board » à condition que vous n'utilisiez pas plus de 2 servomoteurs et qu'ils soient de petite puissance – en cas contraire vous risquez d'endommager la platine).

Saisissez ensuite le programme présenté ci-après (ce dernier est disponible sur notre site Internet : www.lextronic.fr ou sur notre CD-ROM sous le nom « smcpro »).

La petite application vous permettra de modifier indépendamment la position (0 à 180°) de 2 servomoteurs à l'aide des potentiomètres 1 et 3 de la platine. Les servomoteurs sont reliés sur les sorties 0 et 7 du module de commande « SMCPRO ». Les potentiomètres 2 et 4 permettront quant à eux de modifier indépendamment la vitesse de déplacement des 2 servomoteurs. Il vous sera ainsi très facilement possible de vérifier que les mouvements des 2 servos sont totalement indépendants (en position et en vitesse).

Le programme débute par l'acquisition de la valeur de la tension présente sur le curseur du potentiomètre N° 2. Valeur que l'on divise par 4 pour obtenir un nombre compris entre 0 et 255 (ce paramètre envoyé par voie série au module « SMCPRO » correspondra à la vitesse de déplacement du servomoteur N° 1). Le programme continu avec par l'acquisition de la valeur de la tension présente sur le curseur du potentiomètre N° 2.

Cette valeur (pouvant être comprise entre 0 et 1023) est divisée par 5.68 pour obtenir une variation comprise alors entre 0 et 180.

Ce paramètre correspond à l'angle que devra prendre le palonnier du servomoteur (après qu'on lui ait envoyé l'information par voie série). Les mêmes opérations s'exécutent ensuite pour le servomoteur N° 2 avec les 2 autres potentiomètres.

```
'###################################
'#    Gestion platine « SMCPRO »          #
'#    @Lextronic 2006 - 11/03/2006        #
'###################################

Const Device = CB220

Dim vitesse As Integer
Dim position As Integer
Dim calcul As Single

Opencom 1,9600,3,5,50                          ' Configure le port de communication

Input 0                                        ' Configure les ports en entrée
Input 1
Input 2
Input 3

'****************** Boucle principale *************

Do
  vitesse = Adin(1)/4                          ' Récupère valeur 0 - 255 du potentiomètre 2
  Putstr 1,0,0,vitesse                         ' Envoi l'ordre de vitesse du servo 0
  Delay 10
  calcul = Adin(0) / 5.68                      ' Conversion position potentiomètre -> 0 à 180
  position = calcul
  Putstr 1,0,1,position                         ' Envoi la position du servomoteur
  Delay 10

  vitesse = Adin(3)/4                          ' Récupère valeur 0 - 255 du potentiomètre 4
  Putstr 1,7,0,vitesse                         ' Envoi l'ordre de vitesse du servo 7
  Delay 10
  calcul = Adin(2) / 5.68                      ' Conversion position potentiomètre -> 0 à 180
  position = calcul
  Putstr 1,7,1,position                         ' Envoi la position du servomoteur
  Delay 10
Loop
```

NOTE D'APPLICATION # 23.
Gestion d'un module horloge « RTCBoard »

Cette note d'application va vous permettre de piloter un petit module « RTCBoard ». Ce dernier est une horloge temps réel à base de circuit « DS1302 ». Vous apprendrez ainsi comment programmer l'heure et la date dans le module, mais aussi comment activer la charge de sa capacité de sauvegarde et enfin comment récupérer les données afin de les afficher sur un écran LCD.

Le composant « DS1302 » s'interface au moyen d'un bus 3 fils. Ce dernier dispose entre autre de 7 adresses mémoires dans lesquelles vous pourrez stocker les données relatives aux heures, minutes, secondes, mois , jour, année... (La valeur de ces adresses diffère selon que vous veniez "lire" ou "écrire" dans le composant - voir corespondance avec le tableau ci-dessous).

	En écriture	**En lecture**	**Type de données**
Adresse	&H80	&H81	Secondes
Adresse	&H82	&H83	Minutes
Adresse	&H84	&H85	Heures
Adresse	&H86	&H87	Date Jour
Adresse	&H88	&H89	Mois
Adresse	&H8A	&H8B	Jour semaine
Adresse	&H8C	&H8D	Année

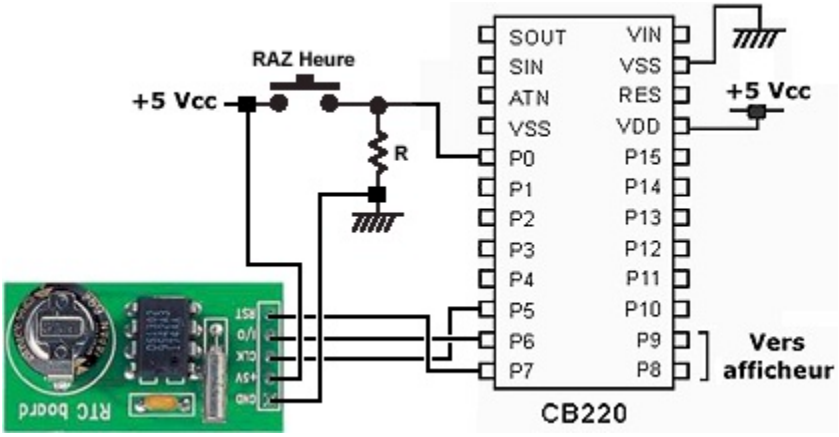
Ainsi à la mise sous tension du montage, si vous appuyez sur un bouton-poussoir (relié sur le port P0), toutes les données de l'horloge (heure + date) seront initialisées à des valeurs précises et affichées sur l'écran LCD. Si par contre, vous n'appuyez pas sur le bouton-poussoir lors de la mise sous tension du montage, l'heure « en cours » mémorisée dans l'horloge temps réel grâce à la capacité de sauvegarde s'affichera alors à l'écran.

Notions abordées :
- Gestion SPI™
- Gestion I2C™ pour l'afficheur LCD

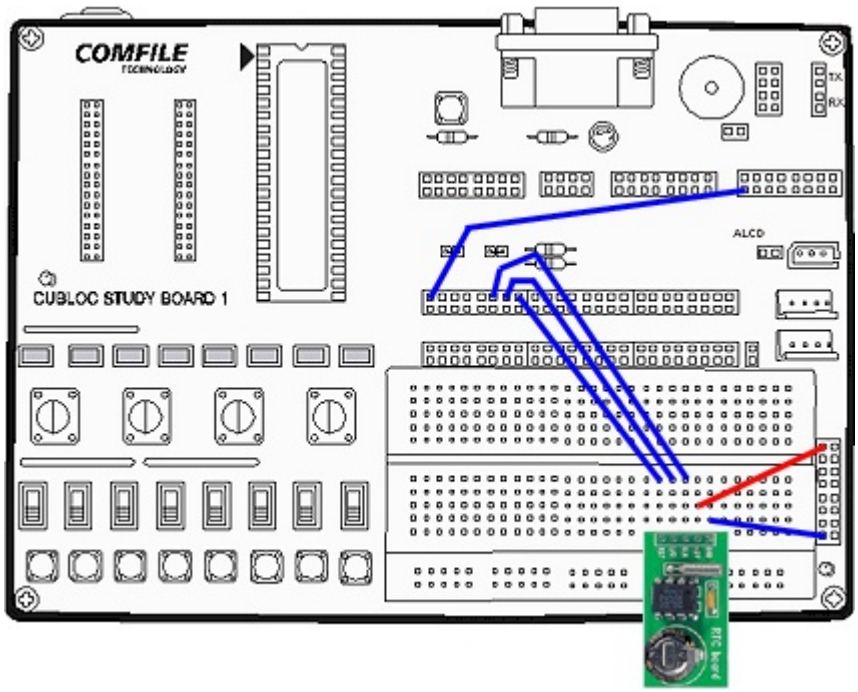
Matériel nécessaire :
- Une platine « CUBLOC Study Board » (facultative)
- 1 module « RTC Board »
- 1 Afficheur LCD Comfile + 1 bouton-poussoir

Préparation matérielle :

Celle-ci repose sur le schéma théorique ci-dessous.



Afin de faciliter la description de cette note d'application, nous utiliserons une platine « CUBLOC Study Board » associée à un module CUBLOC™ CB220 (voir schéma de raccordement ci-après).



Saisissez ensuite le programme présenté ci-après (ce dernier est disponible sur notre site Internet : www.lextronic.fr ou sur notre CD-ROM sous le nom « rtcboard »).

Ce dernier n'apporte pas de commentaire particulier et reste simple à interpréter.

```
'#################################
'#    Gestion platine « SMCPRO »         #
'#    @Lextronic 2006 - 11/03/2006       #
'#################################

Const Device = CB220
Set Display 2,0,1,50
Dim heure As Byte
Dim minute As Byte
Dim seconde As Byte
Dim annee As Byte
Dim mois As Byte
Dim jour As Byte
Low    0                                    ' configure Port 0 en entrée
High     7                                  ' configure port 7 en sortie avec niveau haut
High   5
High     6
Delay 800
Cls
Delay 200
Csroff
Delay 200

' ###################################################################
' # configuration recharge capacite du DS1302                       #
' ###################################################################

  High 7                                    ' Désactive protection du DS1302
  Shiftout 5,6,0,&h8e,8
  Shiftout 5,6,0,0,8
  Low 7
  Delay 10
  High 7
  Shiftout 5,6,0,&h90,8                     ' Active recharge de la super CAPA
  Shiftout 5,6,0,&HA5,8
  Low 7
  Delay 10

  ' ###################################################################
  ' # Mise à l'heure / date du DS1302 si BP appuyé                    #
  ' ###################################################################

  If In(0) = 1 Then
    High 7
    Shiftout 5,6,0,&h80,8                   ' Mise à jour des secondes
    Shiftout 5,6,0,&H15,8                   ' Ici 15 sec.
    Low 7
    Delay 1
    High 7
    Shiftout 5,6,0,&h82,8                   ' Mise à jour des minutes
    Shiftout 5,6,0,&H42,8                   ' Ici 42 mn.
    Low 7
    Delay 1
    High 7
    Shiftout 5,6,0,&h84,8                   ' Mise à jour des heures
    Shiftout 5,6,0,&H19,8                   ' Ici 19 heure.
    Low 7
    Delay 1
    High 7
    Shiftout 5,6,0,&h8C,8                   ' Mise à jour de l'année
    Shiftout 5,6,0,&H06,8                   ' Ici 2006.
    Low 7
    Delay 1
    High 7
```

```
    Shiftout 5,6,0,&h88,8                                   ' Mise à jour du mois
    Shiftout 5,6,0,&H03,8                                   ' Ici 03.
    Low 7
    Delay 1
    High 7
    Shiftout 5,6,0,&h86,8                                   ' Mise à jour du jour
    Shiftout 5,6,0,&H11,8                                   ' Ici 11.
End If

' ################################################################
' # récupération affichage de l'heure                            #
' ################################################################

Do
   High 7                                                   ' Recuperation des secondes
   Shiftout 5,6,0,&H81,8
   seconde = Shiftin(5,6,4,8)
   Low 7
   Delay 100
   High 7                                                   ' Recuperation des minutes
   Shiftout 5,6,0,&H83,8
   minute = Shiftin(5,6,4,8)
   Low 7
   Delay 100
   High 7                                                   ' Recuperation des heures
   Shiftout 5,6,0,&H85,8
   heure = Shiftin(5,6,4,8)
   Low 7
   Locate 0,0                                               ' Affichage de l'heure complète
   If heure < 9 Then Print "0"                              ' affichage d'un zero si < 9
     Print Hex heure,":"
     If minute < 9 Then Print "0"                           ' affichage d'un zero si < 9
     Print Hex minute,":"
     If seconde < 9 Then Print "0"                          ' affichage d'un zero si < 9
     Print Hex seconde

     ' ################################################################
     ‘ # récupération affichage de la date                          #
     ' ################################################################

      High 7                                                ' Recuperation de l'année
      Shiftout 5,6,0,&H8D,8
      annee = Shiftin(5,6,4,8)
      Low 7
      Delay 10
      High 7                                                ' Recuperation du mois
      Shiftout 5,6,0,&H89,8
      mois = Shiftin(5,6,4,8)
      Low 7
      Delay 10
      High 7                                                ' Recuperation du jour
      Shiftout 5,6,0,&H87,8
      jour = Shiftin(5,6,4,8)
      Low 7
      Locate 10,0                                           ' Affichage de la date complète
      If jour < 9 Then Print "0"                            ' affichage d'un zero si < 9
      Print Hex jour,"/"
      If mois < 9 Then Print "0"                            ' affichage d'un zero si < 9
      Print Hex mois,"/20"
        If annee < 9 Then Print "0"                         ' affichage d'un zero si < 9
        Print Hex annee
        Delay 200
     Loop
```

NOTE D'APPLICATION # 24.
Gestion d'un afficheur LCD « Demmel products »

Cette note d'application va vous permettre de piloter un Afficheur LCD graphique « Demmel products ». Cette gamme d'afficheurs "OEM" avec contrôleur intégré fait partie des modèles parmi les plus performants disponibles actuellement sur le marché. Dotés ou non (suivant les modèles) de touches taclites programmables et d'une résolution de 128 x 64 pixels jusqu'à 320 x 240 pixels, ces derniers pourront être très facilement interfacés avec un ordinateur ou un microcontrôleur externe au moyen de leur multiples interfaces: RS-232, RS-232, I2C™, USB.

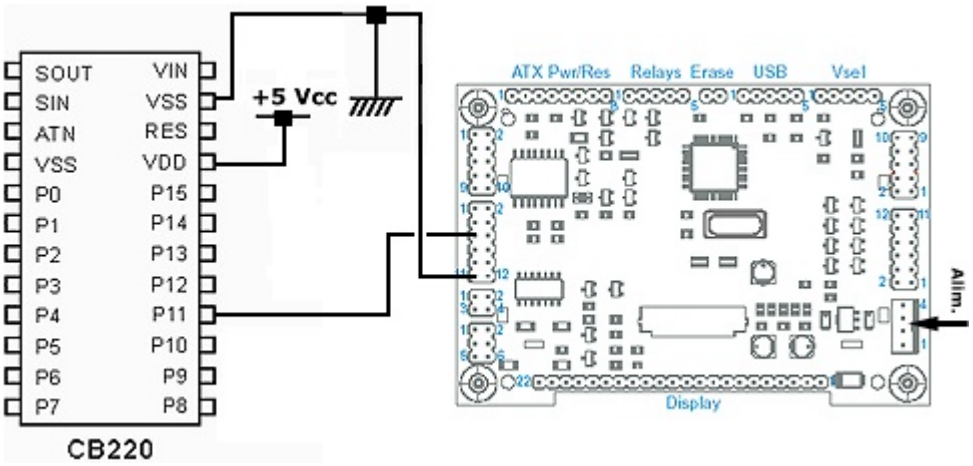
Pour cette note d'application, nous utiliserons le modèle 128 x 64 pixels (sans dalle tactile) qui sera piloté en mode série. Ce dernier bénéficie d'un excellent rapport qualité / prix / performance et d'une grande qualité d'affichage.

Notions abordées :
- Gestion communication série
- Création de « fonctions »

Matériel nécessaire :
- Un « CB220 » + 1 afficheur « Demmel products » 128 x 64 pixels

Préparation matérielle :

Le raccordement à l'afficheur se fera comme indiqué ci-dessous. On alimentera l'afficheur par sa propre source (bloc secteur livré avec le starter-kit de ce dernier).



Saisissez ensuite le programme présenté ci-après (ce dernier est disponible sur notre site Internet : www.lextronic.fr ou sur notre CD-ROM sous le nom « ilcd »).

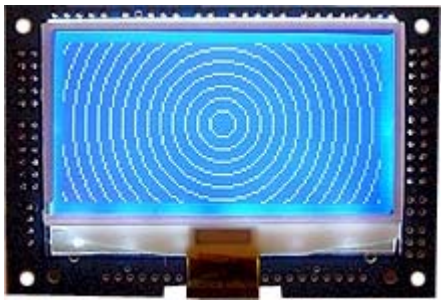
Ce dernier permet en fait de reproduire (via des ordres envoyés par le CUBLOC) un des écrans de présentation pré-chargé dans la « démo » livrée avec chaque afficheur. Cet exemple réutilise également les fontes et certaines images préchargées de la démo (si vous avez donc modifié le contenu de la démo via le logiciel de configuration de l'afficheur, il vous faudra recharger la démo de ce dernier afin d'obtenir les mêmes résultats). Bien que non directement compatible avec les commandes d'affichages spécifiques des CUBLOCS, il vous sera néanmoins très facile de piloter les afficheurs « Demmel products » par l'envoi d'ordres séries successifs. Le programme ci-après indique pour chaque commande série, le résultat recherché sur l'afficheur. Il vous sera donc très facilement possible à l'aide de la notice et du logiciel de configuration de l'afficheur d'en comprendre la signification. Vous apprendrez ainsi à afficher du texte en justifiant ce dernier (centrage à l'écran, centrage à gauche), à afficher du texte avec différentes fontes et différents effets (gras, souligné, vidéo inverse…), à afficher des lignes, des images pré-mémorisées, des rectangles…

Les poses de 50 ms réalisées entre chaque commande sont destinée à laisser à l'afficheur le temps d'exécuter sa commande. Cette durée peut être réduite et optimisée selon la nature de la commande. On utilise cette methode afin de ne pas avoir à tester l'état de l'afficheur et ainsi afin de pouvoir piloter ce dernier uniquement avec un fil de commande TX.

```
'########################################
'#     Gestion afficheur « Demmel products»     #
'#     @Lextronic 2006 - 22/04/2006             #
'########################################

Const Device = CB220

Opencom 1,115200,3,10,70
Pause 1000

Putstr 1,&HAA,"!"                          ' Efface l'écran
Pause 500
Putstr 1,&HAA,"DR",&H19,0,128,0,11         ' Trace un rectangle plein
Pause 50
Putstr 1,&HAA,"AF",0,1                     ' Récupere fonte 1 (de la démo préchargée dans l'afficheur)
Pause 50
Putstr 1,&HAA,"CK",0,0,0,2                 ' Positionne curseur sur LCD (X=0,Y=2)
Pause 50
Putstr 1,&HAA,"AI",1                       ' Mode vidéo inverse
Pause 50
Putstr 1,&HAA,"CT",&H81,0,0,0,0            ' Centre texte sur LCD
Pause 50
Putstr 1,&HAA,"DTCUBLOC <> iLCD",0         ' Affiche texte sur LCD
Pause 50
Putstr 1,&HAA,"AI",0                       ' Mode vidéo normal
Pause 50
Putstr 1,&HAA,"AF",0,0                     ' Récupere fonte 0
Pause 50
Putstr 1,&HAA,"CK",0,0,0,13                ' Positionne curseur sur LCD (X=0,Y=13)
Pause 50
Putstr 1,&HAA,"CT",&H81,0,0,0,0            ' Centre texte sur LCD
```

```
Pause 50
Putstr 1,&HAA,"DTCet exemple montre ",&H1B,&H5B,&H31,&H6D,"quelques",&H1B,&H5B,&H6D," possibilites des
afficheurs ",&H1B,&H5B,&H34,&H6D,&H80,"LCD",0
Pause 50
Putstr 1,&HAA,"CL",&H55                         ' Configure ligne pointillée
Pause 50
Putstr 1,&HAA,"CK",0,0,0,39                     ' Configure point de départ ligne
Pause 50
Putstr 1,&HAA,"DL",0,127,0,39                   ' Trace ligne pointillée
Pause 50
Putstr 1,&HAA,"CK",0,0,0,37                     ' Positionne curseur sur LCD (X=0,Y=37)
Pause 50
Putstr 1,&HAA,"CT",&H81,0,0,0,0                 ' Centre texte sur LCD
Pause 50
Putstr 1,&HAA,"DT",&H1B,&H7B,&H39,&H46,"www.demmel.com",&H1B,&H7B,&H46,0        ' Affiche texte sur LCD
Pause 50
Putstr 1,&HAA,"CK",0,4,0,48                     ' Configure position sur LCD (X=4,Y=48)
Pause 50
Putstr 1,&HAA,"G",0,7                           ' Affiche petite image pré-mémorisé dans la démo (logo ILCD)
Pause 50
Putstr 1,&HAA,"CK",0,49,0,49                    ' Configure position sur LCD (X=49,Y=49)
Pause 50
Putstr 1,&HAA,"CT",&H80,0,0,0,0                 ' Formate texte à gauche sur LCD
Pause 50
Putstr 1,&HAA,"DTNext generation Intelligent LCDs",0          ' Affiche texte sur LCD
Do
Loop
```

Nous vous proposons une seconde petite note d'application dans laquelle nous avons recours à des fonctions écrites pour simplifier le pilotage de certaines commandes de l'afficheur.
Saisissez le programme présenté ci-après (ce dernier est disponible sur notre site Internet : www.lextronic.fr ou sur notre CD-ROM sous le nom « ilcd2 »).



Ce programme va par exemple dessiner successivement :

- Plusieurs rectangles de tailles et de styles différents à l'écran (plein, vide, avec ombrage, avec coins arrondis…).
- Plusieurs style de lignes (pointillées, pleines, diagonales…)
- Plusieurs cercles concentriques.

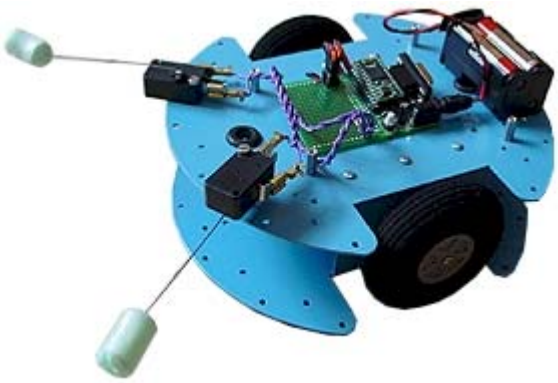
Libre à vous alors de créer vos propres fonctions afin de pouvoir exploiter trèsfacilement les possibilités des afficheurs « Demmel products ».

```
'########################################
'#     Gestion afficheur « Demmel products»     #
'#     @Lextronic 2006 - 22/04/2006             #
'########################################

Const Device = CB220

Dim a As Byte
Dim b As Byte
Dim c As Byte

Opencom 1,115200,3,10,70
Pause 1000
Do

 '***************** Test tracés de rectangles *************
 b=5
 c=5
 For a = 0 To &H1F                          ' Test les différents styles de rectangles
  Putstr 1,&HAA,"!"                         ' Efface l'écran
  Pause 500
  iLCD_BOX a,0,b,0,c                        ' Trace un rectangle avec le style défini par variable "a"
  Pause 500
  b=b+1                                     ' La taille du rectangle augmente au fur et à mesure
  c=c+1
 Next

 '********************** Test tracés de lignes *************

 Putstr 1,&HAA,"!"                          ' Efface l'écran
 iLCD_STLINE &HFF                           ' Sélectionne ligne pleine
 iLCD_LINE 127,0                            ' Trace la Ligne

 iLCD_SET 0,5                               ' Position le pointeur plus bas sur le LCD
 iLCD_STLINE &H55                           ' Sélectionne ligne pointillée
 iLCD_LINE 127,5                            ' Trace la Ligne

 iLCD_SET 0,10                              ' Position le pointeur plus bas sur le LCD
 iLCD_STLINE &HF0                           ' Sélectionne ligne pointillée (autre style)
 iLCD_LINE 127,10                           ' Trace la Ligne

 iLCD_SET 0,15                              ' Position le pointeur plus bas sur le LCD
 iLCD_STLINE &H33                           ' Sélectionne ligne pointillée (autre style)
 iLCD_LINE 127,15                           ' Trace la Ligne

 iLCD_SET 0,20                              ' Position le pointeur plus bas sur le LCD
 iLCD_STLINE &H27                           ' Sélectionne ligne pointillée (autre style)
 iLCD_LINE 127,20                           ' Trace la Ligne

 iLCD_STLINE &HFF                           ' Sélectionne ligne pleine
 For a = 0 To 127 Step 4
    iLCD_SET 64,30                          ' Position le pointeur au centre en bas du LCD
    iLCD_LINE a,62                          ' Trace la Ligne
 Next
 Pause 5000                                 ' Le rayon du cercle augment au fur et a mesure


 '********************** Test tracés de cercles *************

Putstr 1,&HAA,"!"                           ' Efface l'écran
iLCD_SET 64,32                              ' Pointe le centre de l'écran LCD
For a = 5 To 100 Step 5
    iLCD_CIRCLE a                           ' Trace un cercle sur le LCD
Next                                        ' Le rayon du cercle augment au fur et a mesure
```

```
  Pause 5000
Loop
End


'############################
'# Sous Routine iLCD_BOX    #
'############################

 Sub iLCD_BOX(mode As Byte,x1 As Byte,x2 As Byte,y1 As Byte,y2 As Byte)
    Putstr 1,&HAA,"DR",mode,x1,x2,y1,y2
    Delay 50
 End Sub

'############################
'# Sous Routine iLCD_SET    #
'############################

Sub iLCD_SET(x1 As Byte,y1 As Byte)
    Putstr 1,&HAA,"CK",0,x1,0,y1
    Delay 50
End Sub

'############################
'# Sous Routine iLCD_CIRCLE #
'############################

Sub iLCD_CIRCLE(rayon As Byte)
    Putstr 1,&HAA,"DC",0,rayon
    Delay 50
End Sub

'############################
'# Sous Routine iLCD_STLINE #
'############################

Sub iLCD_STLINE(st As Byte)
    Putstr 1,&HAA,"CL",st
    Delay 50
End Sub

'############################
'# Sous Routine iLCD_LINE   #
'############################

Sub iLCD_LINE(x1 As Byte, y1 As Byte)
    Putstr 1,&HAA,"DL",0,x1,0,y1
    Delay 50
End Sub
```

NOTE D'APPLICATION # 25.
Gestion d'une base robotique « Rogue Blue »

Cette note d'application va vous permettre de piloter une petite base robotique « Rogue Blue » à partir d'un « CB220 ». Cette base est composée d'un support aluminium dont la motricité est assurée par 2 servomoteurs type modélisme, lesquels ont été modifiés afin de pouvoir disposer d'une rotation dans les 2 sens sans butée. Ainsi, lorsque l'on envoi un signal PWM sur le servomoteur afin que ce dernier se positionne normalement au neutre, le servomoteur arrêtera de tourner. Lorsque vous envoyez un signal PWM pour que ce dernier se positionne sur +90° il tournera à fond dans un sens et si vous envoyez un signal PWM pour le mettre en position -90°, il tournera à fond dans l'autre sens.



En envoyant des valeurs de signaux PWM intermédiaires, il sera donc possible de sélectionner à la fois la vitesse et le sens de rotation des servomoteurs (idéal pour la gestion du déplacement de votrebase robotique). Le « robot » réalisé dans cette notre d'application sera capable de se déplacer seul dans une pièce en contournant les obstacles qu'il détectera au moyen d'antennes « capteurs ». Ces antennes seront réaliser au moyen de microswitch auxquels ont aura associé des tiges « palpeuses ». Ces capteurs ne sont pas livrés d'origine avec la base « Rogue Blue » - il vous faudra les acquérir séparément.

Attention à impérativement mettre un dispositif de protection au bout des antennes afin d'éviter tout accident et contact avec les yeux (pour vous-même, des petits enfants, des animaux…). Dans un même ordre d'esprit, la base « Rogue Blue » ainsi que l'application décrite **ci-après ne sont pas des jouets**. Il vous faudra donc impérativement les éloigner et ne pas les utiliser près de jeunes enfants de moins de 10 ans.

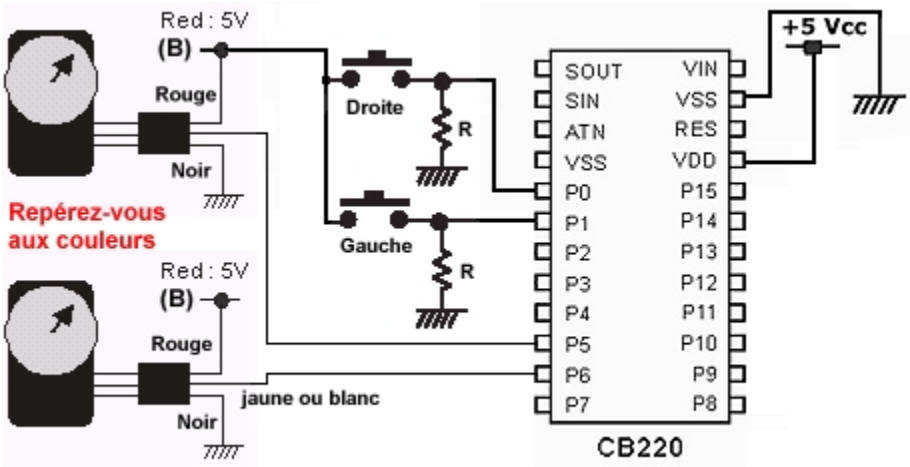
Pour faciliter la réalisation, nous avons utilisé une platine « CB220-Proto » afin que vous n'ayez pas à réaliser votre propre circuit imprimé.

Notions abordées :
- Gestion servomoteur via signaux « PWM »
- Rototique « ludique »

Matériel nécessaire :
- Un « CB220 »
- Une plaque « CB220-Proto »
- 2 microswitch + lige palpeuse + 2 éléments de protection
- 1 base « Rogue Blue »
- 2 supports de piles
- Prise alimentation pour connecteur de la plaque « CB220-Proto »
- Connecteurs et fils divers.

Préparation matérielle :

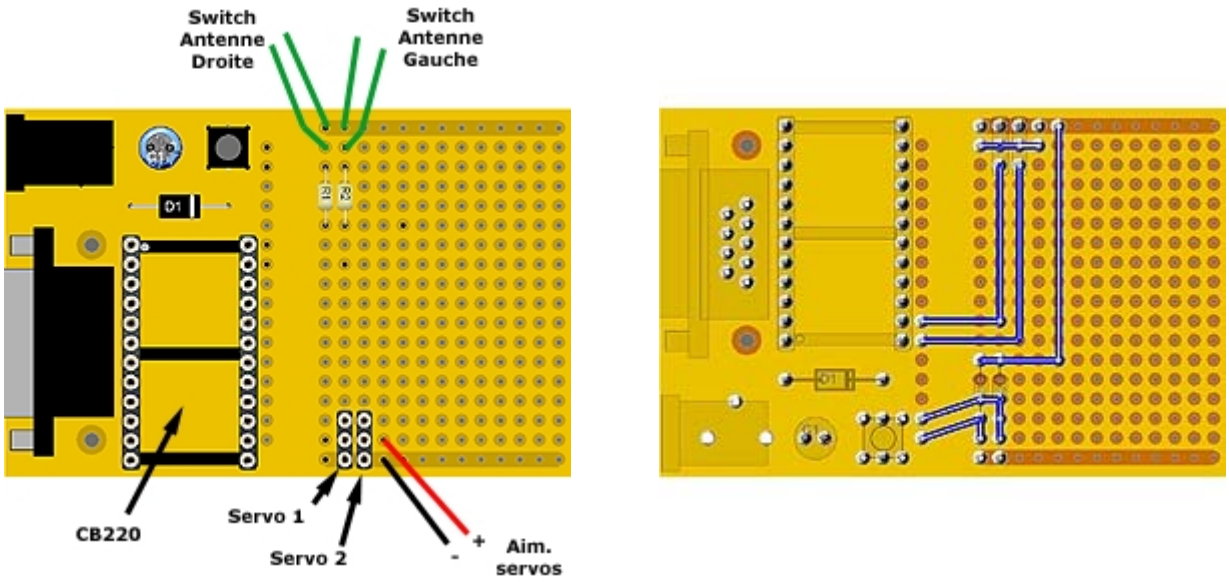
Le schéma théorique du robot est très simple comme indiqué ci-dessous.



L'alimentation des servomoteurs de la base robotique « Rogue Blue » et du module « CB220 » pourra être unique (tension de +5 V) – ce qui est le cas sur ce schéma. Toutefois dans les schémas donnés ci-dessous avec la platine « CB220-Proto », nous avons obté pour 2 alimentations séparées de + 6V (via 1 bloc de 4 x piles 1,5 V) pour le « CB220 » et (via 1 autre bloc de 4 x piles 1,5 V) pour les servomoteurs (le schéma théorique ci-dessus ne correspond donc pas au schéma théorique utilisé avec la platine « CB220-Proto ».

Réalisation à l'aide de la platine « CB220-Proto » :

Le schéma de gauche ci-dessous, montre la platine « CB220-Proto » avec les connexions vers les 2 antennes palpeuses (microswitch). Les résistances R1 et R2 sont de valeur 10 Kohms.



Vue de dessus **Vue de dessous**

On remarque aussi l'alimentation séparée de la platine (via le connecteur en haut à gauche et des servomoteurs fils rouge et noir en bas à droite). On remarque aussi les 2 connecteurs mâles 3 points au pas de 2,54 mm qu'il vous faudra souder sur la platine pour pouvoir recevoir les servomoteurs. A ce titre, vérifiez impérativement la polarité de branchement de vos servomoteurs lorsque vous enfichez les connecteurs sur la platine (repérez-vous par rapport à l'alimentation des servomoteurs). Vous disposez également de la vue du dessous de la platine avec les connexions à réaliser.

Une fois le montage réalisé et vérifié, saisissez ensuite le programme présenté ci-après (ce dernier est disponible sur notre site Internet : www.lextronic.fr ou sur notre CD-ROM sous le nom « rblue »).

La compréhension de ce dernier est très aisée. La boucle principale consiste à vérifier si une ou l'autre des antennes a été sollicitée (via le test des entrées P0 et P1). Si tel n'est pas le cas, on initialise la variable « a » avec la valeur 0. Si en revanche une ou deux antennes sont sollicitées, on envoi dans un premier temps un signal PWM sur les 2 sermoteurs pour les mettre en position neutre (pour arrêter les moteurs). Après une petite temporisation nécessaire pour être sur que la base s'est bien stoppée, on va vérifier l'état exact des antennes et associer une valeur à la variable « a » suivant les différents cas de figure possible (a = 1 pour l'antenne gauche, a = 2 pour l'antenne droite et a = 3 si les 2 antennes sont sollicitées) en même temps. Puis la base effectue une petite marche arrière pour s'éloigner dans un premier temps de l'obstacle rencontré.

Le programme va ensuite en fonction de la valeur de la variable « a » déterminer ce que doit faire la base robotique.

Si a = 0 -> On actionne les 2 servoteurs « marche avant » (aucun obstacle n'avait été détecté).

Si a = 1 -> La base tourne légèrement à droite et le programme continu son exécution (comme il n'y a plus d'obstacle sur les antennes, la base recommencera alors à avancer tout droit).

Si a = 2 -> La base tourne légèrement à gauche et le programme continu son exécution (comme il n'y a plus d'obstacle sur les antennes, la base recommencera alors à avancer tout droit).

Si a = 3 -> La base fait un demi-tour complet sur elle-même et le programme continu son Exécution (comme il n'y a plus d'obstacle sur les antennes, la base recommencera Alors à avancer tout droit).

A noter que les valeurs indiquées dans le programme pour la génération des signaux PWM peuvent être amenées à être légèrement modifiées par vos soins pour fonctionner correctement sur votre application (pour que par exemple les 2 servomoteurs tournent à la même vitesse afin que la base avance bien droit ou que les servomoteurs arrêtent complètement de tourner si vous désirez stopper la base) – Ceci est du à la tolérance de chaque servomoteur.

Libre maintenant à vous de modifier la vitesse de déplacement de la base, de lui ajouter des capteurs de distance ultrasons, une boussole, un module de commande radio par PC (vous avez toutes les informations nécessaires dans les notes d'applications de ce document pour y parvenir !).

```
'##########################################
'#    Gestion d'une base robotique « Rogue Blue»    #
'#    @Lextronic 2006 - 22/04/2006                  #
'##########################################

Const Device = CB220
Dim a As Byte

Input 0                          ' Configure les ports en entrée pour recuperation état des antennes
Input 1
Low 5                            ' Configure les port P5 et P6 en sortie
Low 6
Delay 1000

Do
   If In(0)=1 Or In(1)=1 Then    ' Test si une des 2 antennes n'a pas été sollicitée ?
      Pwm 1,3430,32768           ' Stoppe le robot
      Pwm 0,3405,32768
      Pause 800
      If In(0) = 1 Then          ' Test si antenne gauche sollicitée ?
        a = 1
      Else
        a = 2                    ' C'est que c'est alors l'antenne droite !
       End If
       If In(0) = 1 And In(1)= 1 Then a=3     ' Test si antenne gauche ET droite sollicitée ?
       Pwm 1,3590,32768                       ' Marche arriere
       Pwm 0,3195,32768
       Pause 2500
   Else
      a = 0                                   ‘ Aucune antenne sollicitée -> Le robot avance
   End If

   ' ********** Déplacement du robot en fonction des evenement **********

   Select Case a
        Case 0
            Pwm 0,3590,32768                 ' Marche avant
            Pwm 1,3195,32768
        Case 1                               ' Arrivée sur obstable à gauche
            Pwm 0,3600,32768                 ' Demi-tours léger à droite
            Pwm 1,3600,32768
            Pause 2000
        Case 2                               ' Arrivée sur obstable à droite
             Pwm 1,3180,32768                ‘ Demi-tours léger à gauche
             Pwm 0,3180,32768
             Pause 2000
       Case 3                                ' Arrivée frontale sur un obstable
             Pwm 0,3750,32768                ' Demi-tours complet
             Pwm 1,3750,32768
             Pause 3500
       End Select
Loop
```

NOTE D'APPLICATION # 26.
Gestion d'une base robotique « Araignée »

Cette note d'application va vous permettre de piloter une petite base robotique de type « araignée » à partir d'un « CB220 ». Cette base est composée d'un support plastique avec 6 pattes dont la motricité est assurée par 3 servomoteurs type modélisme. En envoyant différentes valeurs de signaux PWM aux servomoteurs, il vous sera possible de faire déplacer l'araignée de plusieurs façons différentes. Dans le cadre de cette petite note d'application, le programme vous permettra d'enchaîner des mouvements en boucle les uns derrière les autres. L'araignée commence par se redresser, puis se met à sautiller de façon agressive avant d'avancer un moment, de stopper, puis de pivoter vers l'arrière avant de se remettre à sautiller et d'avancer à nouveau et ainsi de suite.

Nous n'avons pas doté la base de capteur permettant de détecter les obstacles (il vous faudra donc placer l'araignée dans un espace dégagé). Toutefois, il vous sera facilement possible de lui ajouter des capteurs de distance ultrasons, une boussole, un module de commande radio par PC, etc… (vous avez toutes les informations nécessaires dans les notes d'applications de ce document pour y parvenir !).

Attention, la base robotique de type « araignée » ainsi que l'application décrite **ci-après ne sont pas des jouets**. Il vous faudra donc impérativement les éloigner et ne pas les utiliser près de jeunes enfants de moins de 10 ans qui pourraient avaler ou inhaler les petits objets qui la composent.

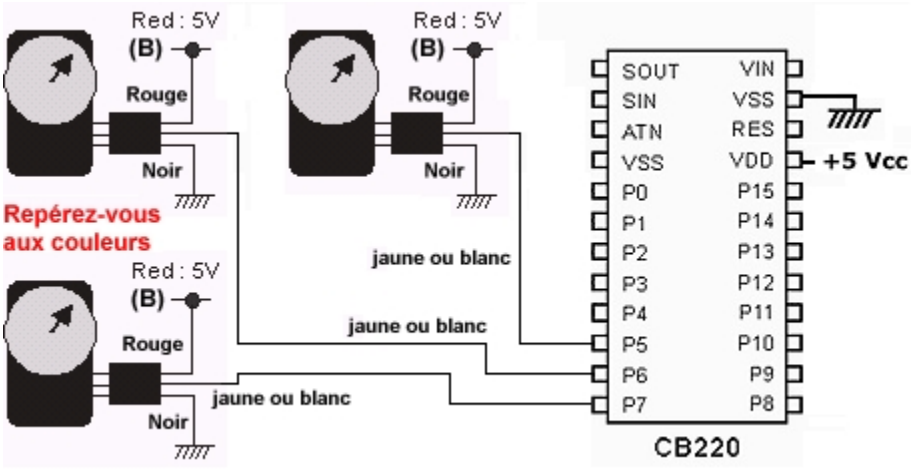
Pour faciliter la réalisation, nous avons utilisé une platine « CB220-Proto » afin que vous n'ayez pas à réaliser votre propre circuit imprimé.

Notions abordées :
- Gestion servomoteur via signaux « PWM »
- Rototique « ludique »

Matériel nécessaire :
- Un « CB220 »
- Une plaque « CB220-Proto »
- 1 base « Rogue Blue »
- 2 supports de piles
- Prise alimentation pour connecteur de la plaque « CB220-Proto »
- Connecteurs et fils divers.

Préparation matérielle :

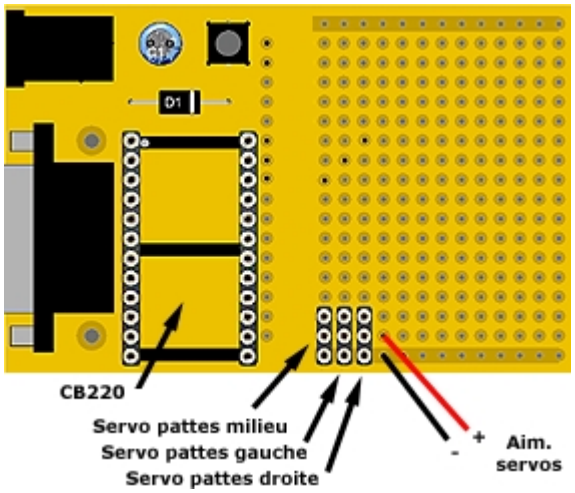
Le schéma théorique du robot est très simple comme indiqué ci-dessous.



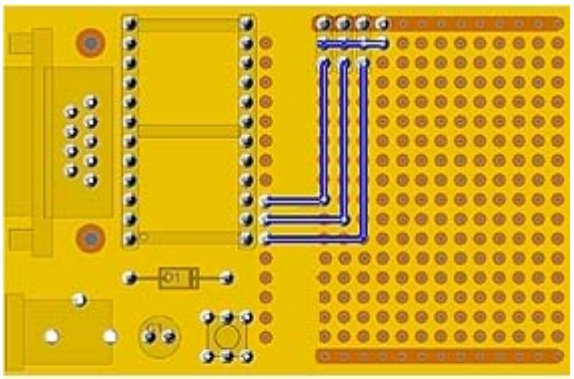
L'alimentation des servomoteurs de la base robotique « Araignée » et du module « CB220 » pourra être unique (tension de +5 V) – ce qui est le cas sur ce schéma. Toutefois dans les schémas donnés ci-dessous avec la platine « CB220-Proto », nous avons obté pour 2 alimentations séparées de + 6V (via 1 bloc de 4 x piles 1,5 V) pour le « CB220 » et (via 1 autre bloc de 4 x piles 1,5 V) pour les servomoteurs (le schéma théorique ci-dessus ne correspond donc pas au schéma théorique utilisé avec la platine « CB220-Proto ».

Réalisation à l'aide de la platine « CB220-Proto » :

Le schéma de gauche ci-dessous, montre la platine « CB220-Proto » avec les connexions vers les 3 servomoteurs.



Vue de dessus



Vue de dessous

On remarque aussi l'alimentation séparée de la platine (via le connecteur en haut à gauche et des servomoteurs fils rouge et noir en bas à droite). On remarque aussi les 3 connecteurs mâles 3 points au pas de 2,54 mm qu'il vous faudra souder sur la platine pour pouvoir recevoir les servomoteurs. A ce titre, vérifiez impérativement la polarité de branchement de vos servomoteurs lorsque vous enfichez les connecteurs sur la platine (repérez-vous par rapport à l'alimentation des servomoteurs). Vous disposez également de la vue du dessous de la platine avec les connexions à réaliser.

Une fois le montage réalisé et vérifié, saisissez ensuite le programme présenté ci-après (ce dernier est disponible sur notre site Internet : www.lextronic.fr ou sur notre CD-ROM sous le nom « spider »).

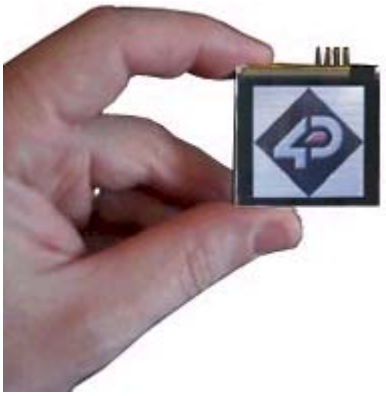
La compréhension de ce dernier est très aisée. Le programme principal consiste à appeler les différentes routines (liées aux mouvements) en boucle. Les mouvements sont réalisé en activant les différents servomoteurs dans leurs positions extrêmes de façon synchronisée. A noter que les valeurs indiquées dans le programme pour la génération des signaux PWM peuvent être amenées à être légèrement modifiées par vos soins pour fonctionner correctement sur votre application (vérifiez surtout que les servomoteurs ne soient pas en buté et ne forcent pas).

```
'###############################################
'#    Gestion d'une base robotique « Araignée »      #
'#    @Lextronic 2006 - 22/04/2006                   #
'###############################################

Const Device = CB220
Dim a As Byte

Low 5                                   ' Configure les port P5, P6 et P7 en sortie
Low 6
Low 7

Do
  Gosub stop                            ' Araignée fixe
  Gosub dance                           ' Araignée dance
  Gosub avance                          ' Arraignée avance
  Gosub stop                            ' Araignée fixe
  Gosub demitours                       ' Fait demi_tours
Loop


' ************** Routine de dance ******************
dance:
   For a = 1 To 5
      Pwm 0,3243,32768                  ' Leve patte mediane gauche
      Pwm 1,3255,32768
      Pwm 2,4253,32768
      Delay 500
      Pwm 0,3243,32768                  ' Leve patte mediane droite
      Pwm 1,3255,32768
      Pwm 2,2300,32768
      Delay 500
Next
Return


' ************** Routine de Stoppe ******************
stop:
```

```
    Pwm 0,3243,32768                    ' Araignée fixe
    Pwm 1,3255,32768
    Pwm 2,3270,32768
    Delay 1000
    Return


' ************** Routine avance ******************
avance:
    For a = 1 To 5
        Pwm 0,4250,32768                ' Araignée avance patte droite gauche
        Pwm 1,4250,32768
        Pwm 2,4250,32768
        Delay 500
        Pwm 0,4250,32768                ' Araignée leve patte mediane droite
        Pwm 1,4250,32768
        Pwm 2,2320,32768
        Delay 500
        Pwm 0,2320,32768                ' Araignée avance patte droite gauche
        Pwm 1,2320,32768
        Pwm 2,2320,32768
        Delay 500
        Pwm 0,2320,32768                ' Araignée leve patte mediane gauche
        Pwm 1,2320,32768
        Pwm 2,4250,32768
        Delay 500
    Next
    Return


' ************** Routine de demi-tours ******************
demitours:
    For a = 1 To 5
        Pwm 0,4250,32768                ' Araignée avance patte droite gauche
        Pwm 1,2320,32768
        Pwm 2,4250,32768
        Delay 500
        Pwm 0,4250,32768                ' Araignée leve patte mediane droite
        Pwm 1,2320,32768
        Pwm 2,2320,32768
        Delay 500
        Pwm 0,2320,32768                ' Araignée avance patte droite gauche
        Pwm 1,4250,32768
        Pwm 2,2320,32768
        Delay 500
        Pwm 0,2320,32768                ' Araignée leve patte mediane gauche
        Pwm 1,4250,32768
        Pwm 2,4250,32768
        Delay 500
    Next
    Return
```

NOTE D'APPLICATION # 27.
Gestion d'un afficheur graphique « MicroLCD »

Cette note d'application va vous permettre de piloter un écran LCD subminiature couleur « microLCD » lequel dispose d'une résolution de 128 x 128 pixels pouvant être affichés en 65536 couleurs ! Il s'interface très facilement au moyen d'une liaison série (niveau 0 / 3 V).

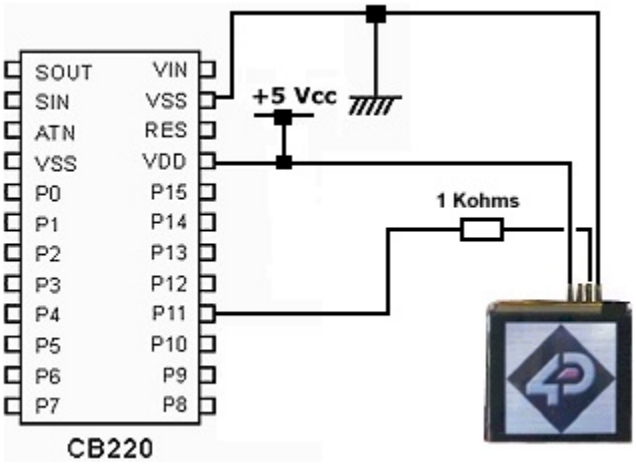


Disponible en 2 versions, ces afficheurs disposent de fonctions diverses qui vous permettrons d'afficher du texte (avec plusieurs tailles de fontes), des graphismes (lignes, cercles, pixels), des images fixes ainsi que de petites animations (mini-vidéo - clip) préalablement chargées au sein de sa mémoire Flash.

Notions abordées :
- Gestion liaison série

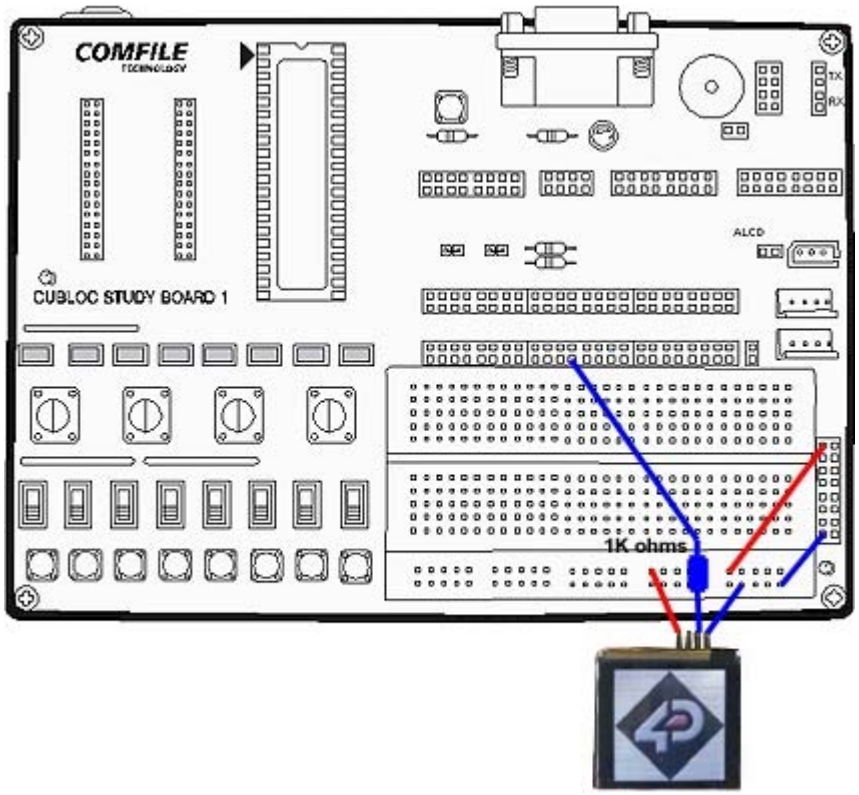
Matériel nécessaire :
- Un « CB220 »
- Un afficheur « Micro-LCD I » ou « Micro-LCD II »

Préparation matérielle :
Le schéma théorique est très simple comme indiqué ci-dessous.



La liaison série de l'afficheur « micro-LCD » est au niveau logique 0 – 3 V, il est donc impératif d'intercaler une résistance de 1 Kohms en série avec la sortie TX du module CUBLOC afin d'éviter la destruction du module d'affichage.

Afin de faciliter la description de cette note d'application, nous utiliserons une platine « CUBLOC Study Board » (voir schéma de raccordement ci-après).



Une fois le montage réalisé et vérifié, saisissez ensuite le programme présenté ci-après (ce dernier est disponible sur notre site Internet : www.lextronic.fr ou sur notre CD-ROM sous le nom « microlcd »).

La compréhension de ce dernier est très aisée. Il va vous permettre de tester une partie des possibilités des afficheurs graphiques couleurs « microLCD ». Ces modèles sont disponibles en 2 versions : « Micro-LCD I » et « Micro-LCD II ». La version « Micro-LCD II » dispose de fonctionnalités supplémentaires comme le fait depouvoir afficher des animations et de disposer d'un jeu d'instructions étendues.

Pour cette note d'application, nous utiliserons des commandes de « bases » communes aux 2 versions d'afficheurs afin de pouvoir afficher du texte avec plusieurs fontes, tracer des lignes, des cercles, des rectangles de couleur, des pixels…

Le programme commence par une temporisation (nécessaire à l'initialisation de l'afficheur lors de sa mise sous tension), puis par l'envoi d'un caractère « U » de synchronisation vers l'afficheur. Ce caractère va permettre à l'afficheur de savoir à quel débit vous désirez communiquer avec lui. Ce dernier dispose en fait d'un système d'auto-détection qui via l'envoi de ce caractère lui permettra de se « calibrer » automatiquement au bon débit. Pour notre part, nous avons choisi de faire travailler le CUBLOC à 9600 bds. A noter que l'afficheur renvoi systèmatiquement un caractère d'acquittement une fois qu'il a reçut une commande « valide ». Dans le cadre de cette notre d'application, nous ne vérifierons pas le retour du caractère, mais nous réaliserons une petite temporisation après chaque commande envoyée à l'afficheur.

Toutefois si votre application nécessite des « timings » très rapides, vous pourrez alors utiliser la sortie TX de l'afficheur (en mettant une résistance de 1 Kohms) afin de pouvoir récupérer le caractère d'acquittement de l'afficheur via l'entrée RX (niveau 0 – 5 V) du CUBLOC) et de vous « synchroniser » sur le retour du caractère. Le programme se poursuit ensuite par l'effacement de l'écran, par la sélection d'une couleur de fond noire et par le réglage du contraste de l'afficheur.

2 tests vous permettrons d'afficher tous les caractères ASCII de l'afficheur avec des couleurs « pseudo-aléatoires » et avec 2 fontes de taille différente. La sélection « pseudo » aléatoire des couleurs se fait à l'aide d'une variable de type Integer dont on récupère l'octet de poid faible et de poid fort. Un autre test permet d'afficher des rectangles de couleurs et de dimensions pseudo aléatoires. L'afficheur ayant une résolution de 128 x 128 pixels, on s'arrangera dans le programme pour que les nombres pseudo aléatoires soient inférieurs à 127. D'autre part, pour la génération d'un rectangle, il est nécessaire d'indiquer à l'afficheur la position des coordonnées x1,y1 du point gauche/haut et x2, y2 du point droit/bas qui formera le rectangle. Le programme générant les coordonnées « pseudo » aléatoires prendra donc en compte le respect de ces contraintes de positionnement.

Un autre test permet de tracer un série de lignes aux dimensions, positionnement et couleurs pseudo aléatoire. Un autre tes permet de positionner une série de pixels à des emplacements et avec des couleurs « pseudo » aléatoire. Le dernier test permet de tracer des clercles avec des rayons, des couleurs et des emplacements « pseudo » aléatoires.

Il est important de signaler que l'afficheur nécessite de recevoir une commande spéciale afin de « désactiver » son étage interne de rétro éclairage avant que vous ne coupiez son alimentation. Si vous ne faites pas cela, l'afficheur pourra dans ces cas extrêmes être endommagé et avoir des lignes de pixels qui ne seront plus affichées. Ainsi en fin de test, un commande visant à faire passer le module en mode « power-down » est envoyée à ce dernier (le rétro-éclairage) se coupe. Une temporisation de 4 secondes est alors effectuée. Si vous devez couper l'alimentation de la platine, faite le à ce moment là. Si vous laissez passer cette temporisation, le programme « re-boucle » à nouveau et la « démo » redémarre.

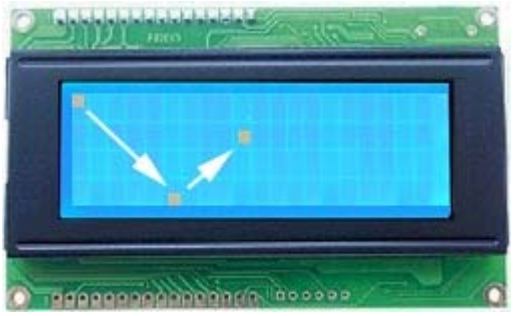
A noter en dernier lieu que l'afficheur nécessite de recevoir le « bon nombre » d'octet lors de l'envoi de ses commandes. Ainsi, lorsque vous commencerez à développer avec ce dernier et à lui envoyer d'autres fonctions, si par exemple vous n'envoyez pas le « bon nombre » d'octet attendu par la fonction, l'afficheur attendra alors de recevoir la suite des données et semblera bloqué. Prenez donc soin de bien suivre la notice de ce dernier.

```
'#############################################
'#    Gestion d'un afficheur « Micro-LCD »   #
'#    @Lextronic 2006 - 10/06/2006           #
'#############################################

Const Device = CB220

        Dim a As Byte
        Dim b As Byte
        Dim c As Byte
        Dim d As Integer
        Dim e As Byte
        Dim f As Byte
        Dim g As Integer
        Dim h As Byte

        Opencom 1,9600,3,10,70
        Pause 3000

        Putstr 1,"U"                    ' Envoi caractère de synchro pour calibration bds
        Pause 500
        Putstr 1,"E"                    ' Efface l'écran
        Pause 500
        Putstr 1,"B",0,0                ' Couleur de fond noir
        Pause 2000
        Putstr 1,"Y",&H02,12            ' Règle le Contraste
        Pause 100

        Do
        ' ********** Test affichage texte fonte 1 ************

        b = 0
        c = 0
        For a = 32 To 127
           d = rnd(0)
           Putstr 1,"T",a,b,c,d.byte0,d.byte1   ' Affiche lettre
           b = b +1
           If b > 20 Then
                    b = 0
             c = c + 1
             If c > 15 Then c = 0
          End If
          Pause 5
        Next
        Pause 3000
        Putstr 1,"E"                    ' Efface l'écran
        Pause 500

        ' ********** Test affichage texte fonte 3 ************

        Putstr 1,"F",2                  ' Sélectionne la fonte 3
        b = 0
        c = 0
        For a = 32 To 127
```

```
    d = rnd(0)
    Putstr 1,"T",a,b,c,d.byte0,d.byte1   ' Affiche lettre
    b = b +1
    If b > 15 Then
              b = 0
      c = c + 1
      If c > 9 Then c = 0
  End If
  Pause 5
Next
Pause 3000
Putstr 1,"E"                              ‘ Efface l'écran
Pause 500

' ********** Test de remplissage surface ************

For a = 0 To 30
    d = rnd(0)
    g = rnd(0)
    b = d.byte0
    c = d.byte1
    d = g.byte0
    e = g.byte1
            If b > 127 Then b = b - 127
    If c > 127 Then c = c - 127
            If d > 127 Then d = d - 127
    If e > 127 Then e = e - 127
    If b > d Then
       h = d
       d = b
       b = h
    End If
            If c > e Then
       h = e
       e = c
       c = h
    End If
    Putstr 1,"p",b,c,d,e,d.byte0,g.byte1     ' Trace surface
    Pause 500
Next
Pause 3000
Putstr 1,"E"                                 ' Efface l'écran
Pause 500

' ********** Test le tracé des lignes ************

For a = 0 To 150
    d = rnd(0)
    g = rnd(0)
    b = d.byte0
    c = d.byte1
    d = g.byte0
    e = g.byte1
            If b > 127 Then b = b - 127
    If c > 127 Then c = c - 127
            If d > 127 Then d = d - 127
    If e > 127 Then e = e - 127
    Putstr 1,"L",b,c,d,e,d.byte0,d.byte1     ' Trace lignes
    Pause 50
Next
Pause 2000
Putstr 1,"E"                                 ' Efface l'écran
Pause 500
```

```
        ' ********** Test le tracé des pixel ************

        For a = 0 To 150
           d = rnd(0)
           b = d.byte0
           c = d.byte1
                 If b > 127 Then b = b - 127
           If c > 127 Then c = c - 127
           Putstr 1,"P",b,c,d.byte0,d.byte1          ' affiche des points
           Pause 50
        Next
        Pause 2000
        Putstr 1,"E"                                 ' Efface l'écran
        Pause 500

        ' ********** Test le tracé des cercles ************
        For a = 0 To 127 Step 2
           d = rnd(0)
           b = d.byte0
           c = d.byte1
                 If b > 127 Then b = b - 127
           If c > 127 Then c = c - 127
                 Putstr 1,"C",b,c,a,d.byte0,d.byte1     ' Trace des cercles
           Pause 200
        Next
        Pause 2000
        Putstr 1,"E"                                  ' Efface l'écran
        Pause 500

        Putstr 1,"Y",&H03,&H00                        ' Mode Power-Down
        Pause 4000                                    ' Coupez l'alimentation... maintenant !
        Putstr 1,"Y",&H03,&H01                        ' Mode Power-Up
        Pause 3500
        Putstr 1,"Y",&H02,12                          ' Règle le Contraste
        Pause 150
Loop
```

NOTE D'APPLICATION # 28.
Conversion d'un afficheur alphanumérique



Cette note d'application va vous permettre de « convertir » un afficheur LCD alphanumérique 4 x 20 caractères en afficheur graphique avec une résolution de 100 x 32 pixels ! Enfin sous cette « accroche » prometteuse, nous allons vous monter qu'il est possible avec certaines limitations, d'utiliser un afficheur LCD alphanumérique pour y afficher le déplacement au pixel prèt d'une petite « balle » via une matrice de 100 x 32 pixels.

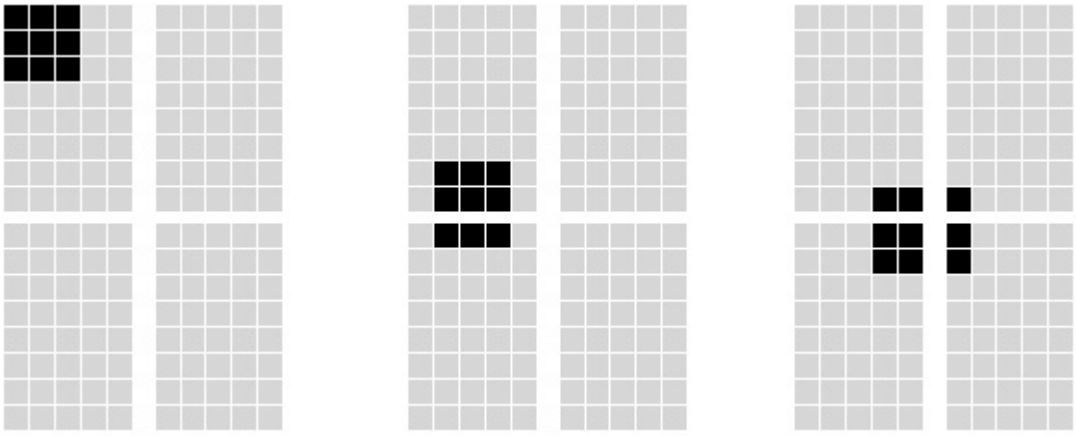
Notions abordées :
- Gestion d'un afficheur LCD

Matériel nécessaire :
- Un « CB220 »
- Un afficheur Comfile Technology 4 x 20 caractères.

Préparation matérielle :

Le schéma théorique est très simple et se limite au raccordement de l'afficheur LCD Comfile technology sur le CUBLOC. Saisissez ensuite le programme présenté ci-après (ce dernier est disponible sur notre site Internet : www.lextronic.fr ou sur notre CD-ROM sous le nom « briques »).

Le principe de cette réalisation repose sur le fait que l'afficheur alphanumérique 4 x 20 caractères dispose en faire de différentes matrices (4 x 20 matrices), lesquelles servent à afficher les caractères. Ces matrices disposent de 5 x 8 pixels. Il est possible en outre de redéfinir jusqu'à 8 caractères en choisissant quels pixels devront être allumés. Pour cette application, nous avons donc décider de redéfinir des caractères afin de pouvoir déplacer une petite « balle » de 3 x 3 pixels sur l'ensemble de la matrice de l'afficheur LCD soit 100 x 32 pixels.



Pour y parvenir il nous a fallu restreindre la surface de « travail » à 4 matrices 5 x 8 pixels, lesquelles permettront de gérer tous les déplacements de la balle en prenant en compte le passage d'une matrice à l'autre. La balle pouvant en effet être affichée sur une seule matrice, 2 ou 4 matrices à la fois suivant sa position.

Les coordonnées de la balle sont mémorisée dans les variables x et y. Ces dernières sont auto-incrémentée / décrémentée pour gérer le déplacement de la balle dans la fenêtre de l'écran et gérer les rebondissemant sur les positions extrêmes. Ces coordonnées sont aussi converties afin de pouvoir connaître la position de la balle sur les 4 matrices reconfigurable ainsi que la position de ses 4 matrices sur l'écran. Le programme consiste alors à utiliser plusieurs variables de données de type « champ à plusieurs dimensions » afin de disposer d'une image « mémoire » des 4 matrices pour y gérer et stocker la position « virtuelle » de chaque point. Une fois cette position mémorisée, le programme dispose de routines qui permettront de redéfinir les 4 caractères en fonction de la position des pixels dans la matrice, pour ensuite afficher les 4 matrices « à la bonne » position sur l'écran LCD. A noter enfin que le programme utiliser en fait 2 jeux de 4 matrices redéfinissables (les 2 jeux de matrice étant utilisés à tours de rôle afin d'éviter des problèmes d'affichage – Un jeu de matrice est modifié hors écran (pendant que l'autre est présent sur le LCD), puis il est affiché à son tours.

Le résultat est alors identique à celui pouvant être réalisé sur un écran LCD graphique (à la limitation de l'espacement qui existe entre les différentes matrices et qui provoque un léger écartement de la balle sur certaines position… mais le résultat est bien là !).

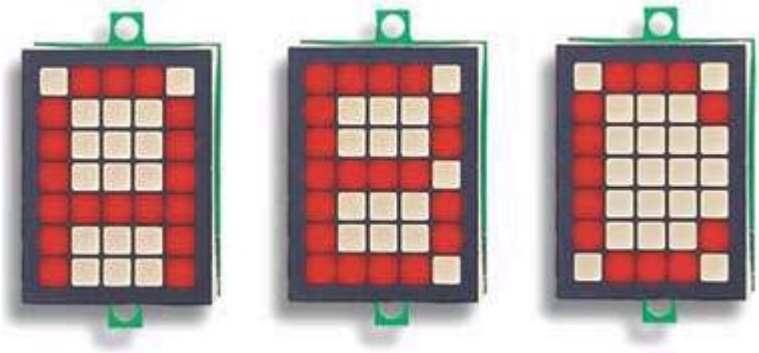
```
'#####################################################
'#    Transformation d'un afficheur « 4 x 20 en 100 x 32 »     #
'#    @Lextronic 2006 - 11/06/2006                             #
'#####################################################

Const Device = CB220                    ' Initialisation de l'afficheur LCD
Set Display 2,0,1,50
Dim m(10,16) As Byte
Dim r(8) As Byte
Dim lcd(20,4) As Byte
Dim x1 As Byte
Dim y1 As Byte
Dim a As Byte
Dim b As Byte
Dim c As Byte
Dim f As Byte
Dim bist As Byte
Dim posx As Byte
Dim posy As Byte
Dim affiche As String*20                ' Gestion affichage
Dim x As Byte
Dim y As Byte
Dim sensx As Byte
Dim sensy As Byte

Delay 500
Cls
Delay 200
Csroff
Delay 100

x = 0
```

```
y = 0
sensx = 0
sensy = 0

Do
  posx = 0
  x1 = x
    Do While x1 > 5
      posx = posx + 1
      x1 = x1 - 6
    Loop
    posy = 0
    y1 = y
    Do While y1 > 8
      posy = posy + 1
      y1 = y1 - 9
    Loop
 ' ************** Redéfinition des caractères en mémoire **************
  For a = 0 To 9
    For b = 0 To 15
      m(a,b) = 0
    Next
  Next

  For a = 0 To 2
    For b = 0 To 2
       m(x1+a,y1+b)=1
     Next
  Next

  For c = 0 To 1
   For a = 0 To 7
      r(a)=0
      For b = 0+(c*5) To 4+(c*5)
        If m(b,a) = 1 Then r(a).bit0=1
        r(a)=r(a)<<1
      Next
         r(a)=r(a)>>1
      Next
   f = 0
      If bist.bit0 = 1 Then f = 4
      Print &H1B,&H44,8+c+f,r(0),r(1),r(2),r(3),r(4),r(5),r(6),r(7)   ' Redéfinition des caractères du LCD
   Next

  For c = 0 To 1
    For a =8 To 15
       f=a-8
       r(f)=0
       For b = 0+(c*5) To 4+(c*5)
        If m(b,a) = 1 Then
           r(f).bit0=1
        End If
        r(f)=r(f)<<1
     Next
    r(f)=r(f)>>1
  Next
  f = 0
  If bist.bit0 = 1 Then f = 4
     Print &H1B,&H44,10+c+f,r(0),r(1),r(2),r(3),r(4),r(5),r(6),r(7)  ' Redéfinition des caractères du LCD
 Next

 ' ************ Raffraichissement de l'écran ************

 For b = 0 To 3
```

```
    For a = 0 To 19
      lcd(a,b)=&h20
    Next
  Next
  If bist.bit0 = 0 Then
    lcd(posx,posy)=8
    lcd(posx+1,posy)=9
    lcd(posx,posy+1)=10
    lcd(posx+1,posy+1)=11
  Else
    lcd(posx,posy)=12
    lcd(posx+1,posy)=13
    lcd(posx,posy+1)=14
    lcd(posx+1,posy+1)=15
  End If

  For b = 0 To 3
    affiche = ""
    For a = 0 To 19
      affiche = affiche + Chr(lcd(a,b))
    Next
    Locate 0,b
    Print affiche
  Next
 bist = bist + 1

 ' ************ Déplacement du point ************
 If sensx = 0 Then
   x = x + 1
   If x > 116 Then
     x = x - 1
     sensx = 1
   End If
 Else
   x = x - 1
   If x =255 Then
     x = 1
     sensx = 0
   End If
 End If

   If sensy = 0 Then
     y = y + 1
     If y > 32 Then
       y = y - 1
       sensy = 1
     End If
   Else
     y = y - 1
     If y = 255 Then
       y = 1
       sensy = 0
     End If
   End If
Loop
```

NOTE D'APPLICATION # 29.
Gestion d'une matrice à Leds « EFN »

Cette note d'application va vous permettre de piloter une matrice à Leds à commande série. Cette matrice composée de 35 Leds se pilote très simplement à partir d'ordres RS-232 (niveau logique + 5V) au moyen de 3 fils (+ / - / signal série) depuis n'importe quel microcontrôleur ou PC (avec interface à base de circuit "MAX-232").



chaque matrice dispose d'un système d'adressage par "ponts de soudure" qui vous permettra de piloter jusqu'à **128 modules** différents ! Une série d'instructions très simples permet d'afficher directement des caractères "ASCII" sur la ma-trice. 2 pattes de fixation sont disponibles de part et d'autre du module.

Octet de poids faible

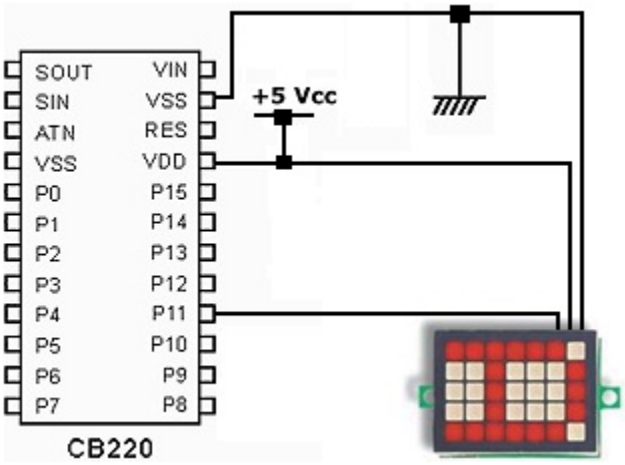
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Octe	2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
ts	3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
de	4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
poid	5	P	Q	R	S	T	U	V	W	X	Y	Z	[	W	]	^	_
s	6	'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
fort	7	p	q	r	s	t	u	v	w	x	y	z			l		

Notions abordées :
- Gestion liaison série

Matériel nécessaire :
- Un « CB220 »
- Un afficheur « EFN »

Préparation matérielle :

Le schéma théorique est très simple et se limite au raccordement de « EFN » sur le CUBLOC. Pour les besoins de cette note d'application, nous n'avons utilisé qu'un seul afficheur « EFN » sur lequel défile un à un les lettres d'un petit texte de 20 caractères. Il vous sera très facilement possible de « chaîner » plusieurs modules « EFN » afin de pouvoir réaliser un véritable panneau défilant.



Saisissez ensuite le programme présenté ci-après (ce dernier est disponible sur notre site Internet : www.lextronic.fr ou sur notre CD-ROM sous le nom « EFN »).

La compréhension du programme est on ne peu plus simple. La phrase à faire afficheur es mémorisée dans une variable de type chaîne (TEXTE). Le programme va prendre les caractères de cette chaîne un par un et les envoyer au module « EFN ».

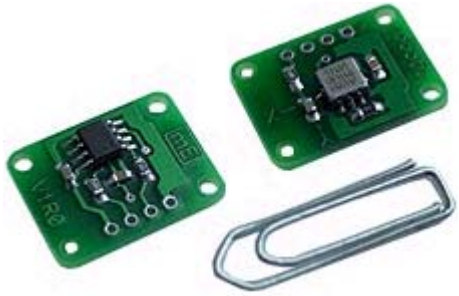
Chaque caractère est précédé des octets 255 et 0 (255 correspond à une commande indiquant au module « EFN » qu'on lui envoi des données à afficher et l'octet « 0 » correspond à l'adresse du module « EFN » (de base les modules sont tous livrés sur la même adresse « 0 »).

Si vous voulez utiliser plusieurs modules « EFN », il vous faudra modifier leur adresse (physiquement sur chaque module) et logiciellement (dans le programme du cubloc parle biais de la valeur de ce 2ème octet).

```
'#######################################################
'#     Gestion d'une matrice à Leds à commande série « EFN »   #
'#     @Lextronic 2006 - 03/07/2006                            #
'####################################################### ##

  Const Device = CB220

  Dim texte As String*20
  Dim b As String*2
  Dim a As Byte

  Opencom 1,9600,3,30,30
  texte="- AFFICHEUR *EFN* - "
  Do
    For a = 0 To 20
      b = Mid(texte,a,1)
      Putstr 1,255,0,b
      Pause 700
    Next
    Pause 2000
  Loop
```

NOTE D'APPLICATION # 30.
Utilisation d'un module accéléromètre « Accel »

Cette note d'application va vous permettre d'interfacer Votre module CUBLOC avec un accéléromètre 2 axes. Ce petit module intègre un composant accélomètre 2 axes (± 2 g) ADXL311 associé à un amplificateur "rail-to-rail". Il pourra être très facilement interfacé à l'aide de ses 2 sorties analogiques afin de pouvoir réaliser des robots ludiques "complexes". Le module sera à ce titre utilisé en tant qu'inclinomètre 2 axes en affichant la position x et y des axes sur un écran LCD.

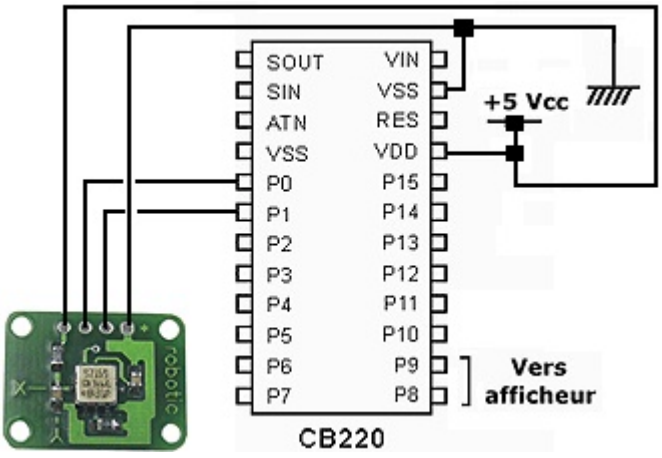


Notions abordées :
- Gestion entrées analogiques

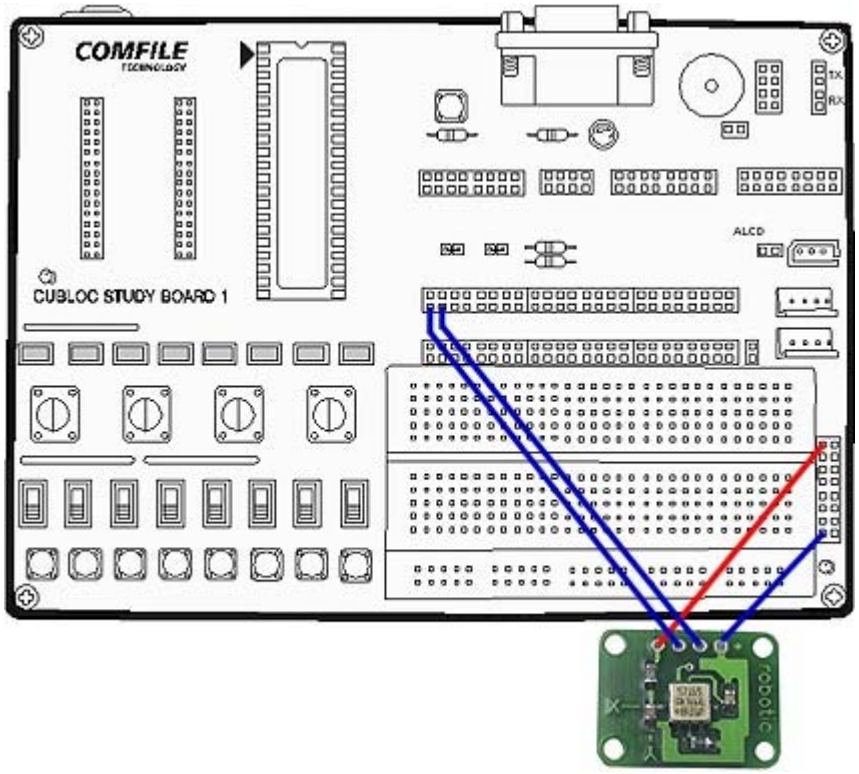
Matériel nécessaire :
- Un « CB220 »
- Un module accéléromètre « Accel »
- Un afficheur « CLCD »

Préparation matérielle :

Le schéma théorique est très simple et se limite au raccordement de « ACCEL » sur le CUBLOC via 2 entrées de conversion analogique / numérique.



Afin de faciliter la description de cette note d'application, nous utiliserons une platine « CUBLOC Study Board » associée à un module CUBLOC™ CB220 (voir schéma de raccordement ci-après).



Saisissez ensuite le programme présenté ci-après (ce dernier est disponible sur notre site Internet : www.lextronic.fr ou sur notre CD-ROM sous le nom « accel »).

La compréhension du programme est on ne peu plus simple. Ce dernier vous permettra d'afficher une indication de l'assiette du capteur. Lorsque ce dernier est à l'horizontal (parallèle au sol), les valeurs des axes X et Y sont à 0. Lorsque vous pencherez le capteur d'un côté ou de l'autre de l'axe horizontal ou vertical, vous obtiendrez un affichage d'une valeur positive ou négative suivant le sens dans lequel vous pencherez le module.

Après avoir récupéré la valeur analogique présente sur chacune des sorties du module « accel », on soustrait une valeur permettant d'obtenir « 0 » au repos (lorsque le capteur est à l'horizontal). Il vous faudra probablement ajuster ces valeurs en fonction de votre capteur.

```
'#################################################
'#     Gestion d'un accéléromètre 2 axes « Accel »          #
'#     @Lextronic 2006 - 03/07/2006                         #
'#################################################

  Const Device = CB220                    ' Initialisations

  Set Display 2,0,1,50
  Dim analog As Long

  Delay 100
  Cls
  Delay 200
  Csroff
  Delay 100

 Input 0                                  ' Configure ports de conversion analogique/numérique en entrée
  Input 1

  Locate 0,0
  Print "Axe X: "
  Locate 0,1
  Print "Axe Y: "

  Do
     analog = Adin(0)                     ' Lecture de la tension du 1 er axe
     analog=analog-510
     Locate 7,0
     Print Dec analog,"   "
     analog = Adin(1)                     ' Lecture de la tension du 2 eme axe
     analog=analog-501
     Locate 7,1
     Print Dec analog,"   "
     Delay 200
  Loop
```