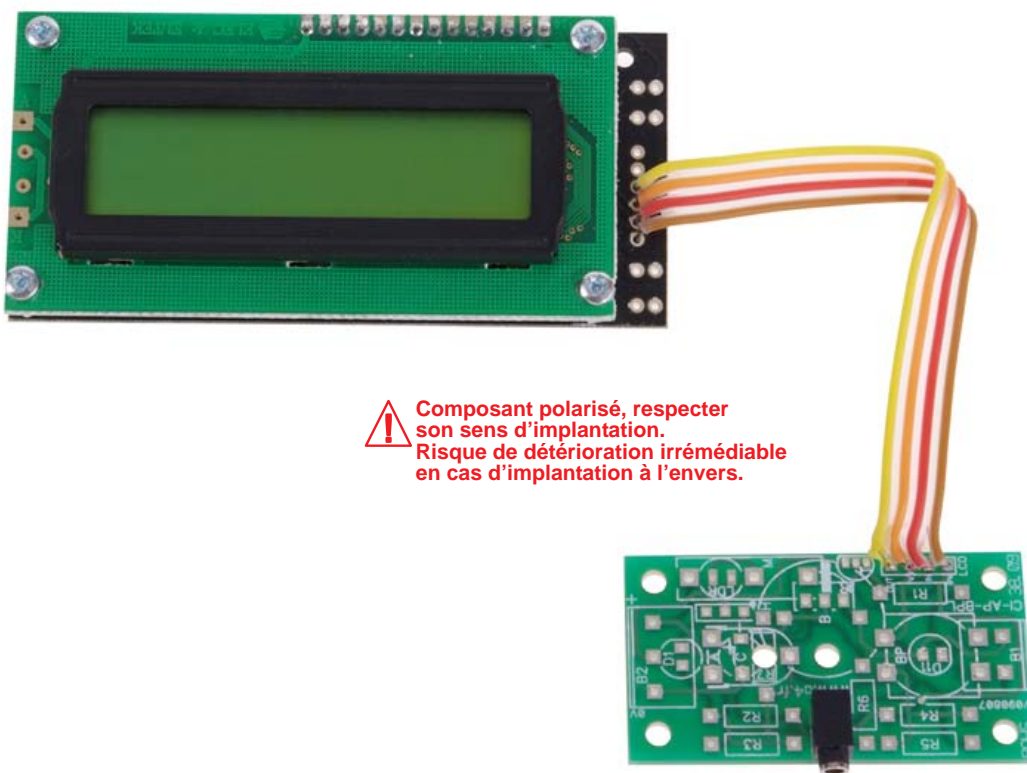


Ecran LCD

Ce module permet d'afficher un message provenant d'un microcontrôleur Picaxe. Il est possible d'afficher jusqu'à 2 lignes de 16 caractères.

La tension d'alimentation de l'afficheur est 4,5V ou bien de 5 à 6V, il est relié à une sortie du microcontrôleur. Le module afficheur décode l'information envoyée par le microcontrôleur et affiche le message paramétré à l'aide de l'instruction "serout".

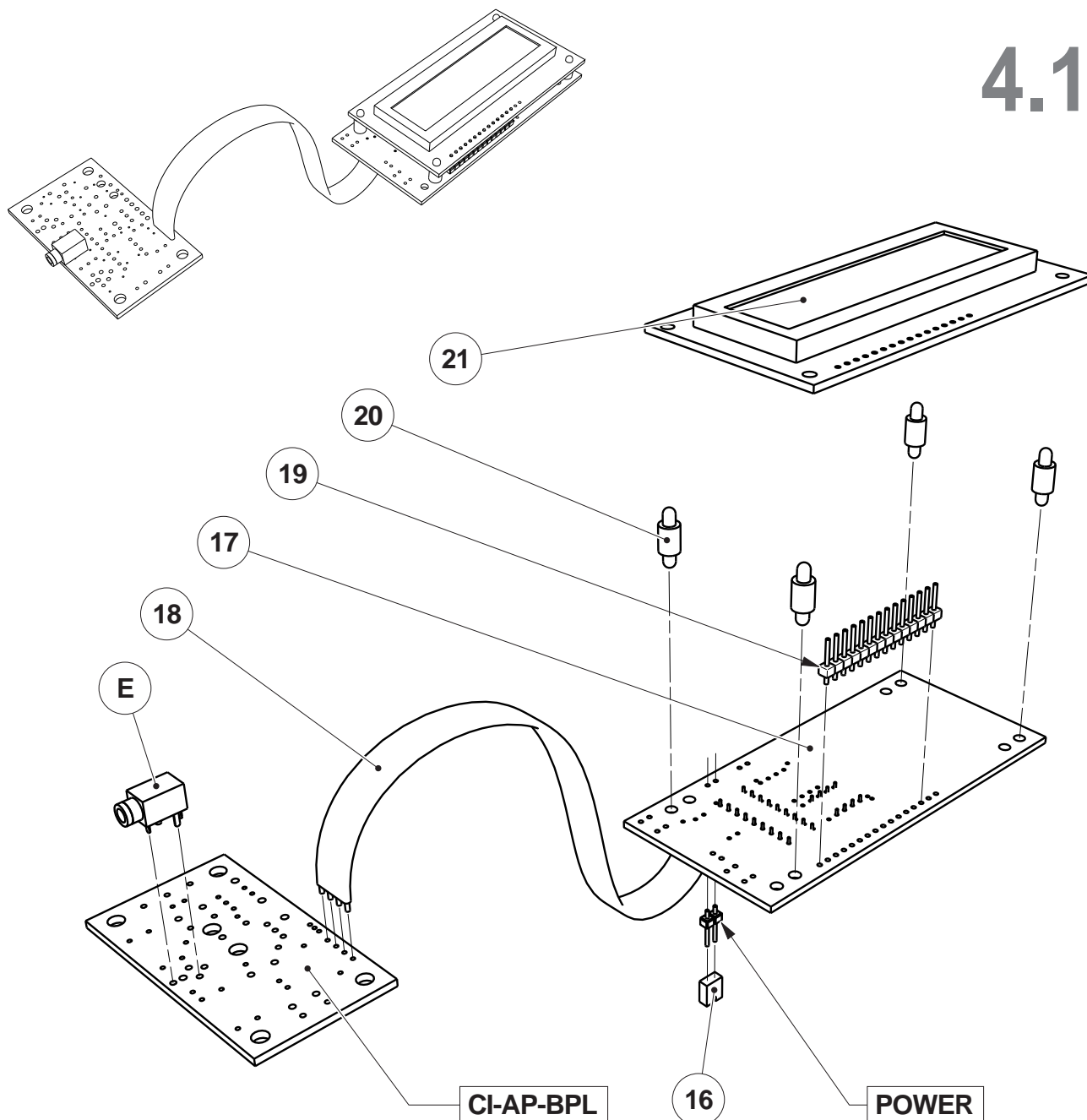


 **Composant polarisé, respecter son sens d'implantation. Risque de détérioration irréversible en cas d'implantation à l'envers.**


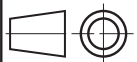
SOMMAIRE

| | |
|-----------------|------------------|
| Nomenclature | 4.11.2 |
| Montage | 4.11.3 |
| Mise en service | 4.11.5 |
| Applications | 4.11.6 à 4.11.14 |

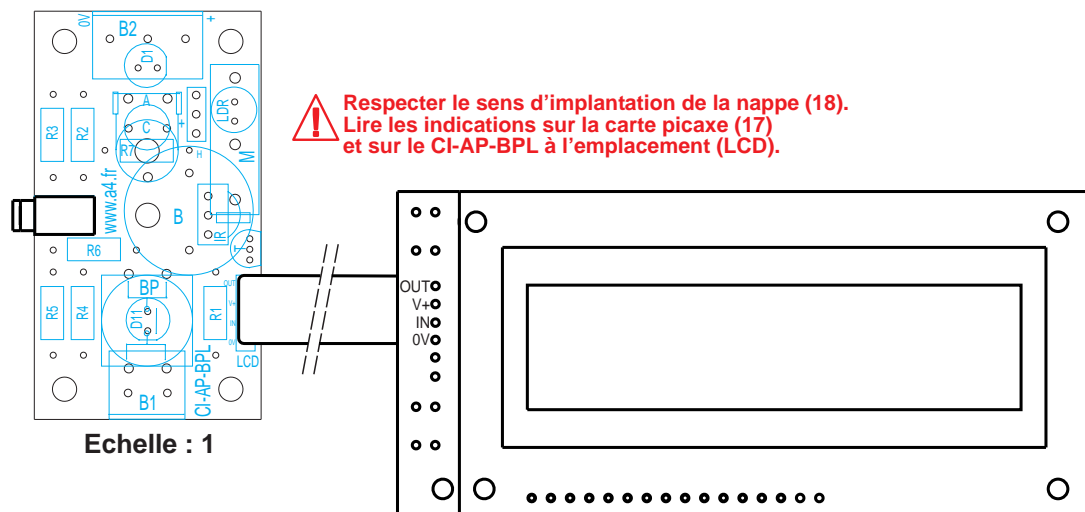
4.11



| | | | |
|-----------|--------|--------------------------------------|--------------------|
| 21 | 01 | Ecran LCD. | |
| 20 | 04 | Entretoise | |
| 19 | 01 | Barrette 14 picots à souder. | |
| 18 | 01 | Nappe de 4 fils. | |
| POWER | 01 | Barrette 2 picots à souder. | |
| 17 | 01 | CI Picaxe LCD. | |
| 16 | 01 | Capuchon | XXXXXXX |
| E | 01 | Embase jack stéréo Ø 2,5 mm pour CI. | EMB-JACK-D2M5A-STE |
| CI-AP-BPL | 01 | Circuit imprimé, 30 x 54. | CI-AP-BPL |
| REPERE | NOMBRE | DESIGNATION | Réf. A4 |

| | | | |
|--|---|------------------------------|-----------------------------------|
|  |  | PROJET | PARTIE |
| | | A4 AutoProg | Module Ecran LCD |
| TITRE DU DOCUMENT | | Nomenclature | |
| Nom | Date | | |

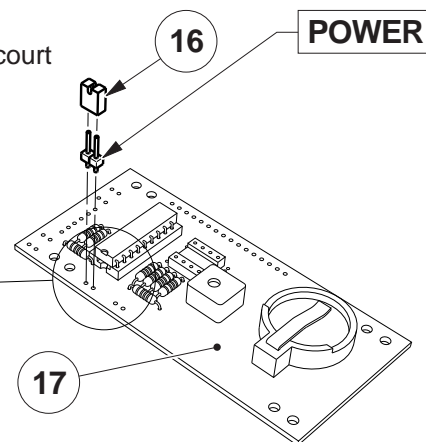
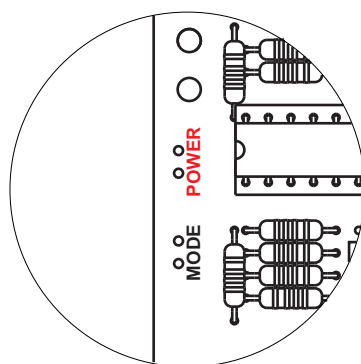
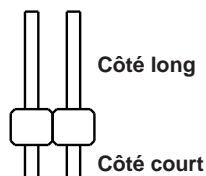
Implantation des composants



Montage de la barrette 2 picots et du capuchon

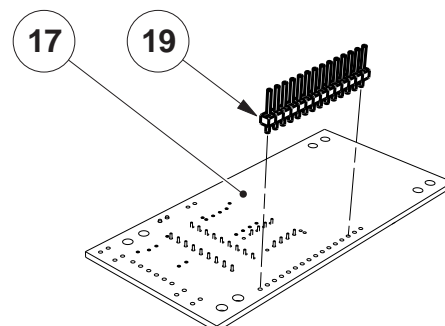
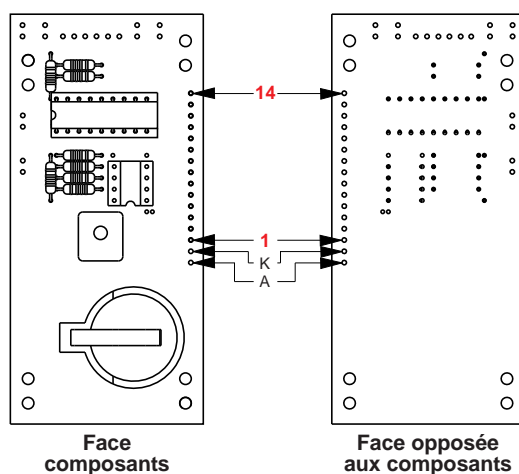
Couper 2 picots sur les barrettes sécables fournies, puis positionner le côté court sur la carte Picaxe (17) à l'emplacement (POWER), et le braser.

Enficher le capuchon (16) dessus.



Montage de la barrette 14 picots

Couper 14 picots sur les barrettes sécables fournies, puis positionner le côté court sur la carte Picaxe (17), face inverse des composants à l'emplacement numéroté 1 14 et braser les 14 picots.

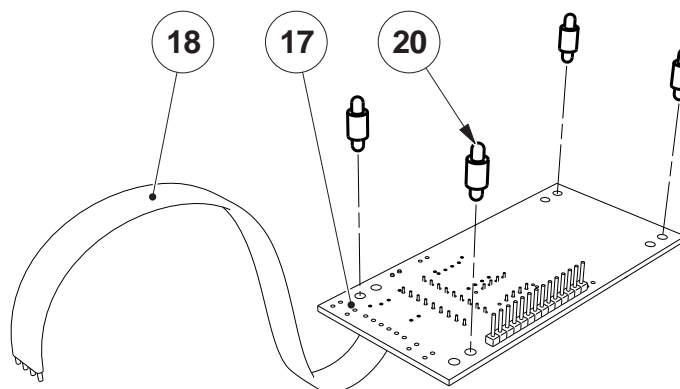
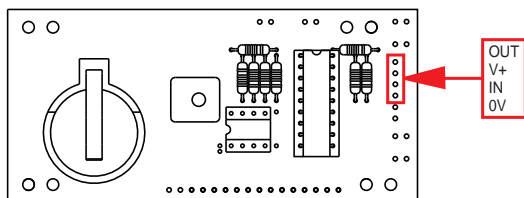


Montage du module écran LCD

Montage des entretoises et de la nappe de 4 fils

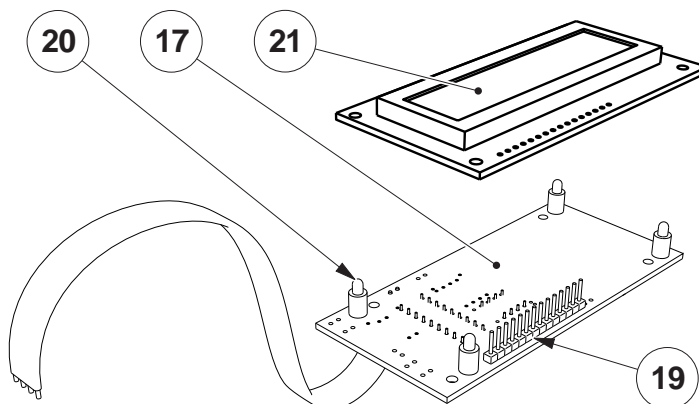
Insérer les 4 entretoises (20) dans la carte Picaxe (17) côté opposé aux composants.
Braser les 4 fils de la nappe (18) sur OUT, V+, IN et 0V.

⚠ Veiller à mettre les 4 entretoises au bon emplacement.



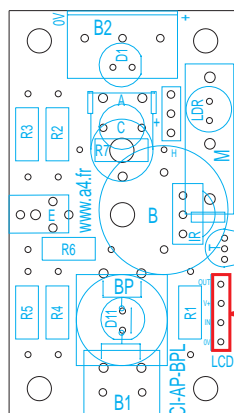
Montage de l'écran LCD

Clipper l'écran LCD (21) sur les entretoises (20) en veillant bien que les picots de la barrette (19) soient bien positionnés dans les trous, puis braser les 14 picots.

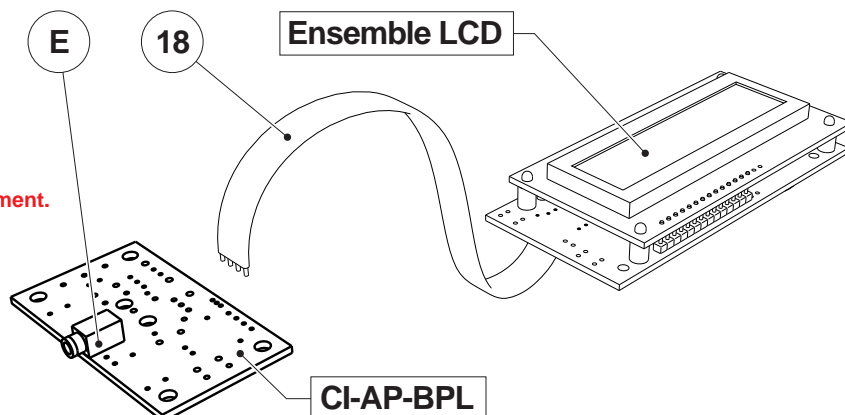


Montage de l'ensemble LCD sur le module

Braser les 4 fils de la nappe (18) sur OUT, V+, IN et 0V du CI-AP-BPL et braser l'embase jack (E).

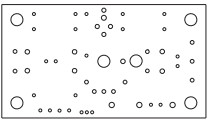

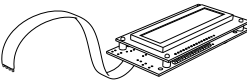


⚠ Veiller à braser la nappe (18) au bon emplacement.



Nomenclature du kit (réf. K-AP-MLCD-M)

Le kit de base comprend toutes les pièces et composants électroniques permettant de réaliser le module LCD.

| Désignation et références A4 | Quantité | Repère | Dessin |
|--------------------------------------|----------|-----------|---|
| Circuit imprimé 30 x 54 x 1,6. | 01 | CI-AP-BPL |  |
| Embase jack stéréo Ø 2,5 mm pour CI. | 01 | E |  |
| Ensemble LCD. | 01 | |  |

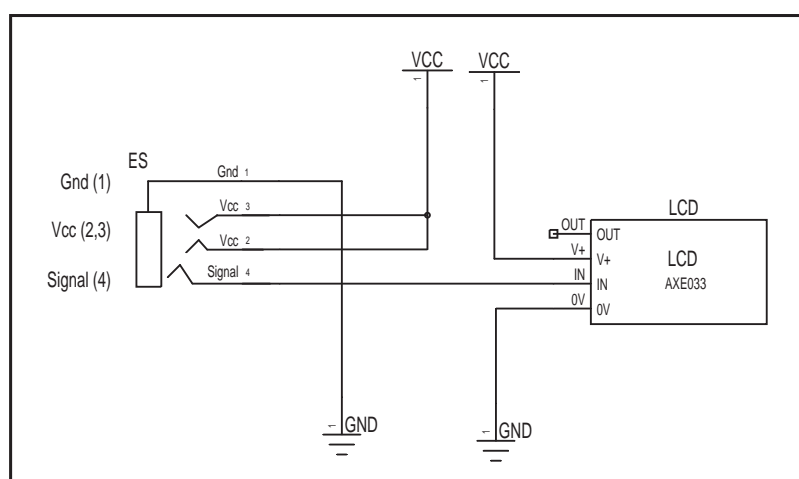


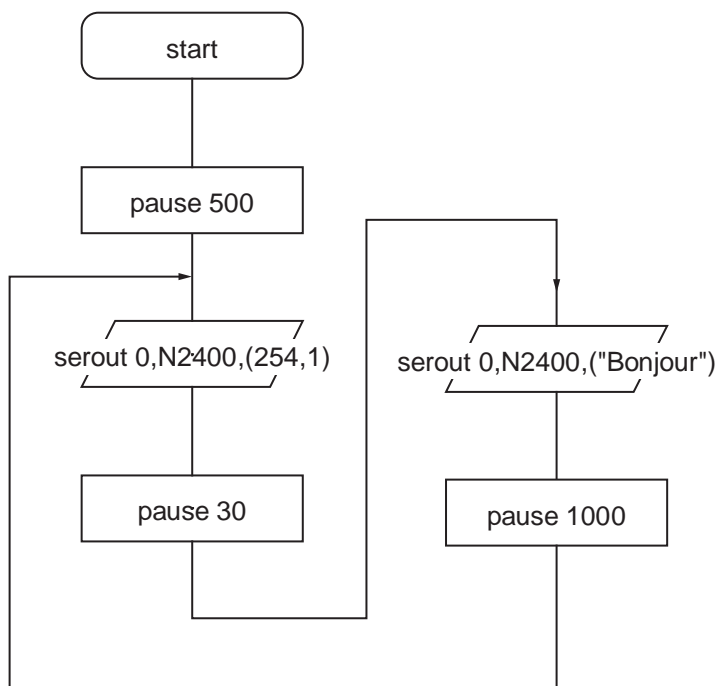
Schéma électronique

Test du module Buzzer

| Phase | Charger le programme nommé | Connecter le module Buzzer sur | Résultats attendus |
|-------|----------------------------|--------------------------------|--|
| 1 | TEST-MLCD.cad | out0 | Tous les points des deux lignes de l'afficheur s'allument et s'éteignent toutes les 0,5 s. |

L'afficheur est connecté sur la sortie Out 0 du microcontrôleur Picaxe. Les programmes suivants sont réalisés dans Programming Editor.

Affichage d'un message fixe sur 1 ligne. Programme : 1-Message 1 ligne.cad



Affichage du mot "Bonjour"

Bonj our

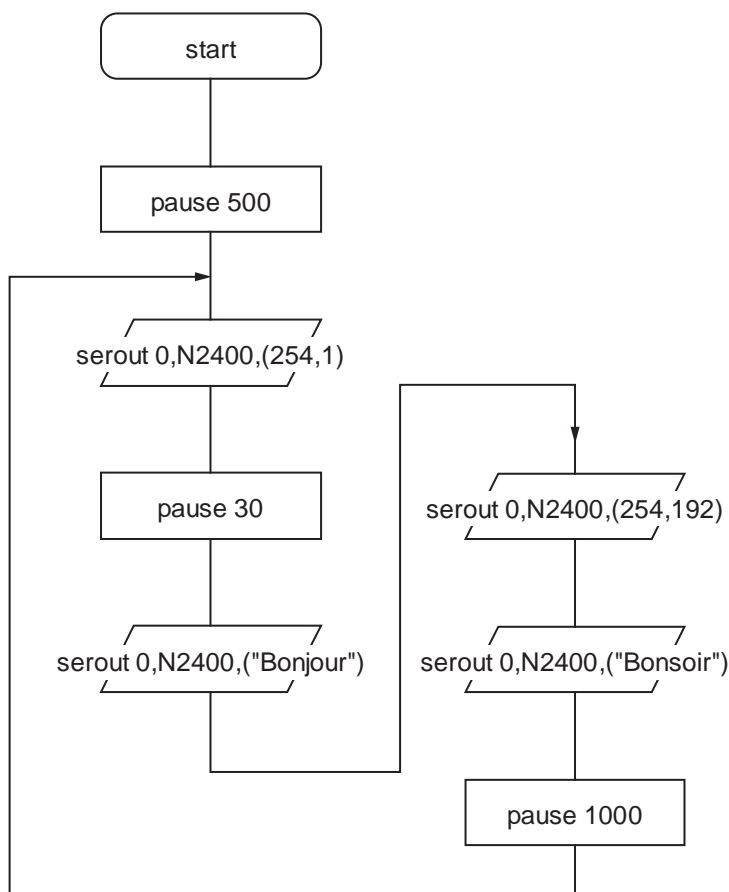
Un temps d'attente de 500 ms est introduit en début de programme pour permettre l'initialisation du module après la mise sous tension.

L'instruction `serout 0, N2400, (254,1)` permet d'effacer le message courant. Un temps d'attente de 30 ms est nécessaire après cette instruction.

L'instruction `serout 0, N2400, ("Bonjour")` permet d'afficher le mot "Bonjour".

Noter qu'un message fixe est encadré par des guillemets "Bonjour".

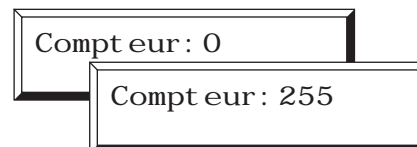
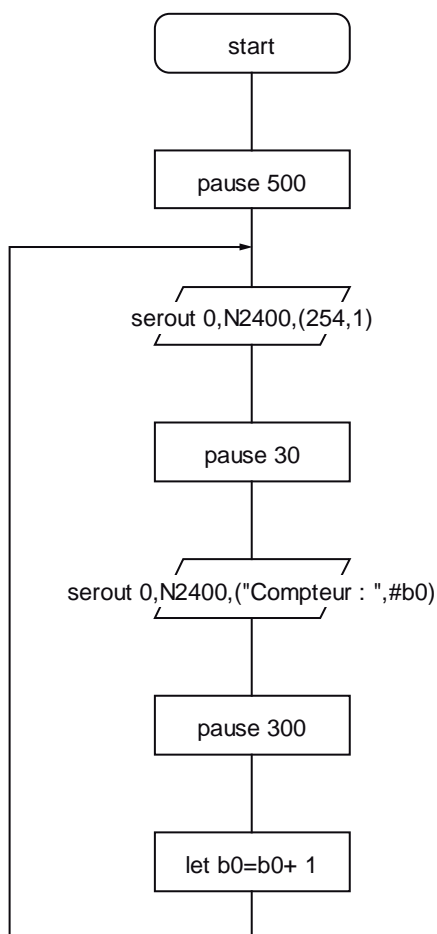
Affichage d'un message fixe sur 2 lignes. Programme : 2-Message 2 lignes.cad



Bonj our
Bonsoi r

L'instruction `serout 0, N2400, (254,192)` permet de sauter une ligne pour afficher le message "Bonsoir".

Affichage d'un message fixe + variable. Programme : 3-Message fixe + variable.cad



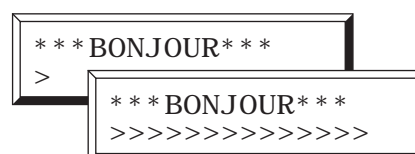
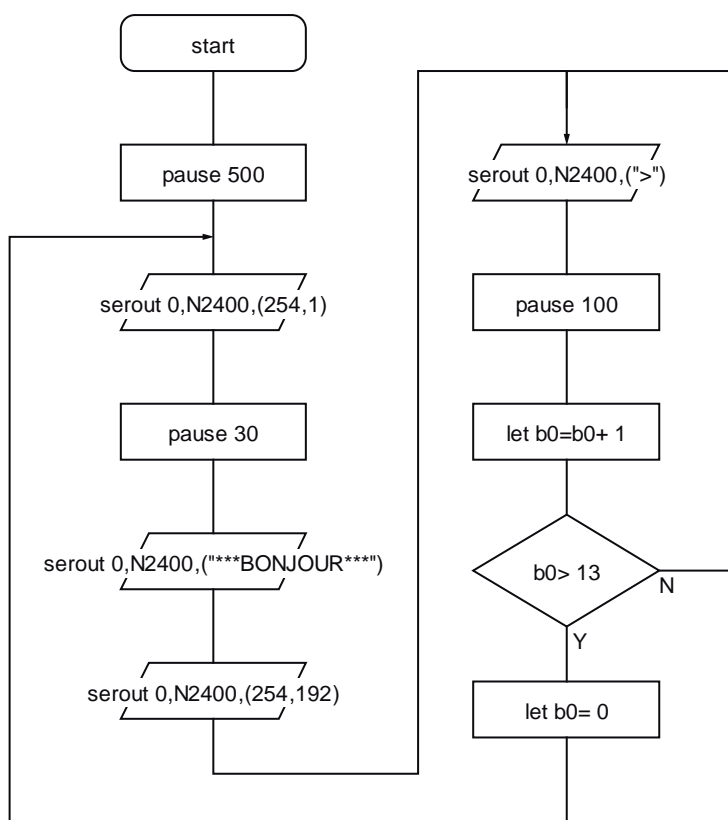
L'instruction serout 0, N2400, ("Compteur : ",#b0) permet d'afficher le message fixe "Compteur" suivi de la valeur contenue dans la variable b0. Ici b0 est incrémentée de 1 en 1 après chaque affichage.

Noter que l'affichage du contenu d'une variable doit être précédé par le symbole dièse (#b0)

La variable b0 pourrait tout aussi bien contenir une valeur provenant d'un capteur (température, humidité, niveau de lumière etc.).
Ex. serait 0, N2400, ("Temp :", #b0, "deg C") afficherait le message Temp : 25 deg C (en supposant que b0=25).

Noter que chaque partie d'un message composé doit être séparé du précédent par une virgule serout 0, N2400, ("Temp : ",#b0, "deg C").

**Affichage d'un message fixe sur la 1^{ère} ligne et animé sur la 2^{ème} ligne.
Programme : 4-Message fixe + animation.cad**



Affichage de *****BONJOUR***** sur la première ligne,
progression du symbole > répété 14 fois
sur la 2^{ème} ligne.

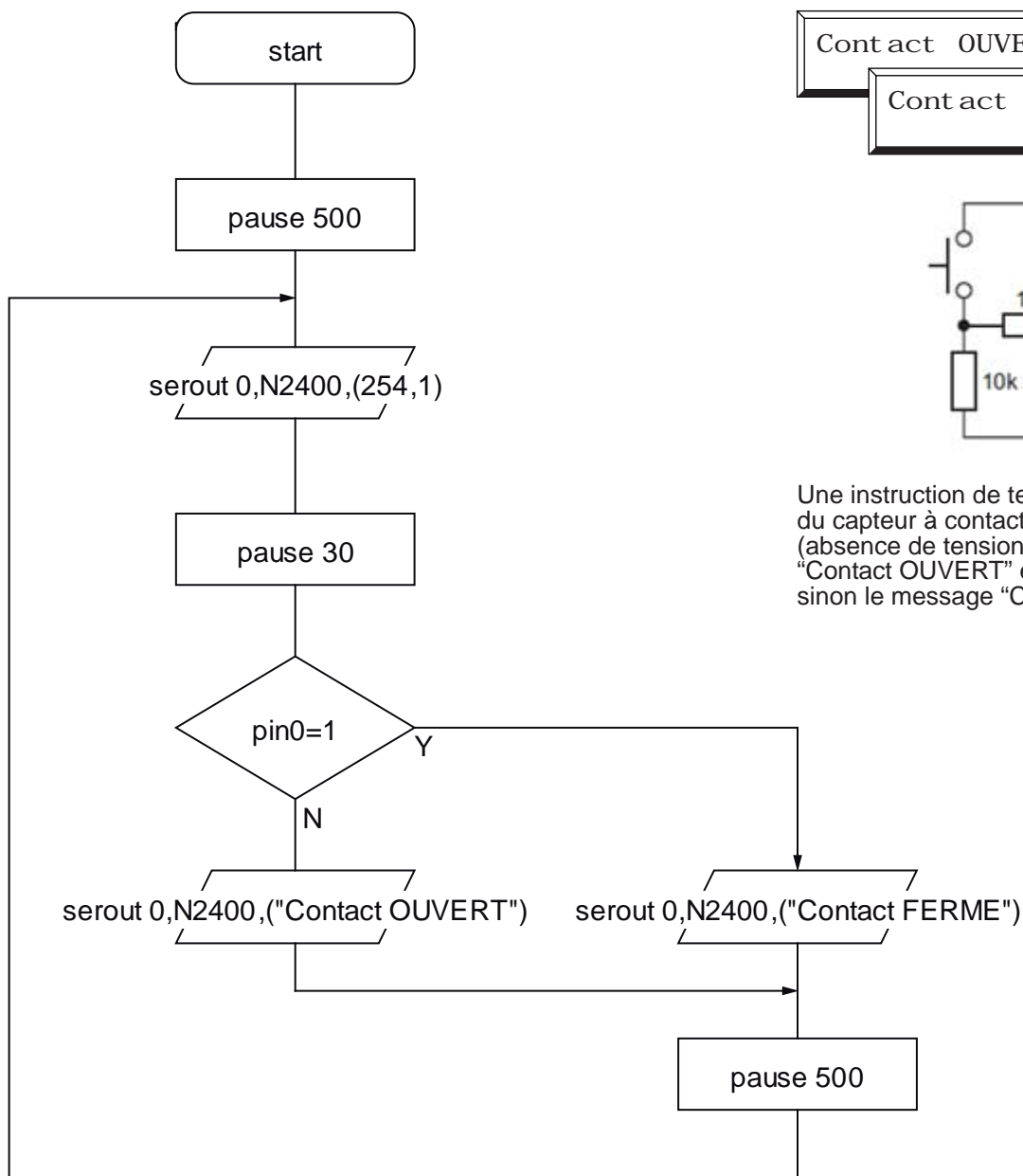
Noter que si l'affichage n'est pas effacé par l'instruction `serout 0, N2400, (254,1)`, le curseur d'affichage se positionne automatiquement sur la position qui suit le message qui vient d'être affiché. Un caractère reste affiché tant que l'affichage n'est pas effacé ou tant qu'il n'est pas écrasé par un nouveau caractère.

Applications avec des capteurs

Les exemples suivants proposent des cas concrets d'applications qui combinent l'utilisation de capteurs avec le module LCD intégré dans un système d'automatisme.

Acquisition et affichage d'un message en fonction de l'état d'un capteur à contact : 5-Message variable avec capteur de contact.cad

Un capteur à contact est connecté sur l'entrée numérique In0 d'un microcontrôleur Picaxe.
On affiche un message qui varie selon que le contact est ouvert ou fermé.

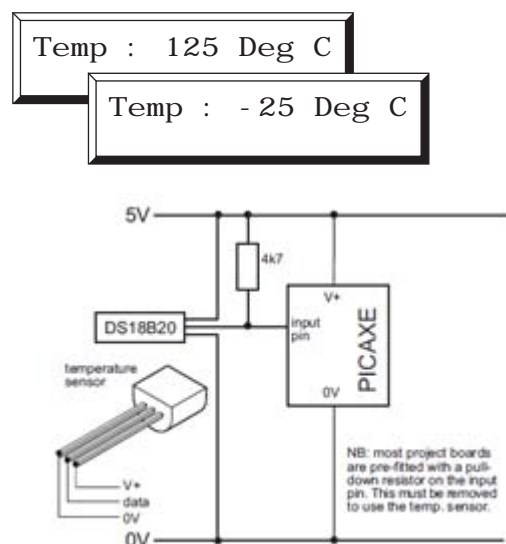
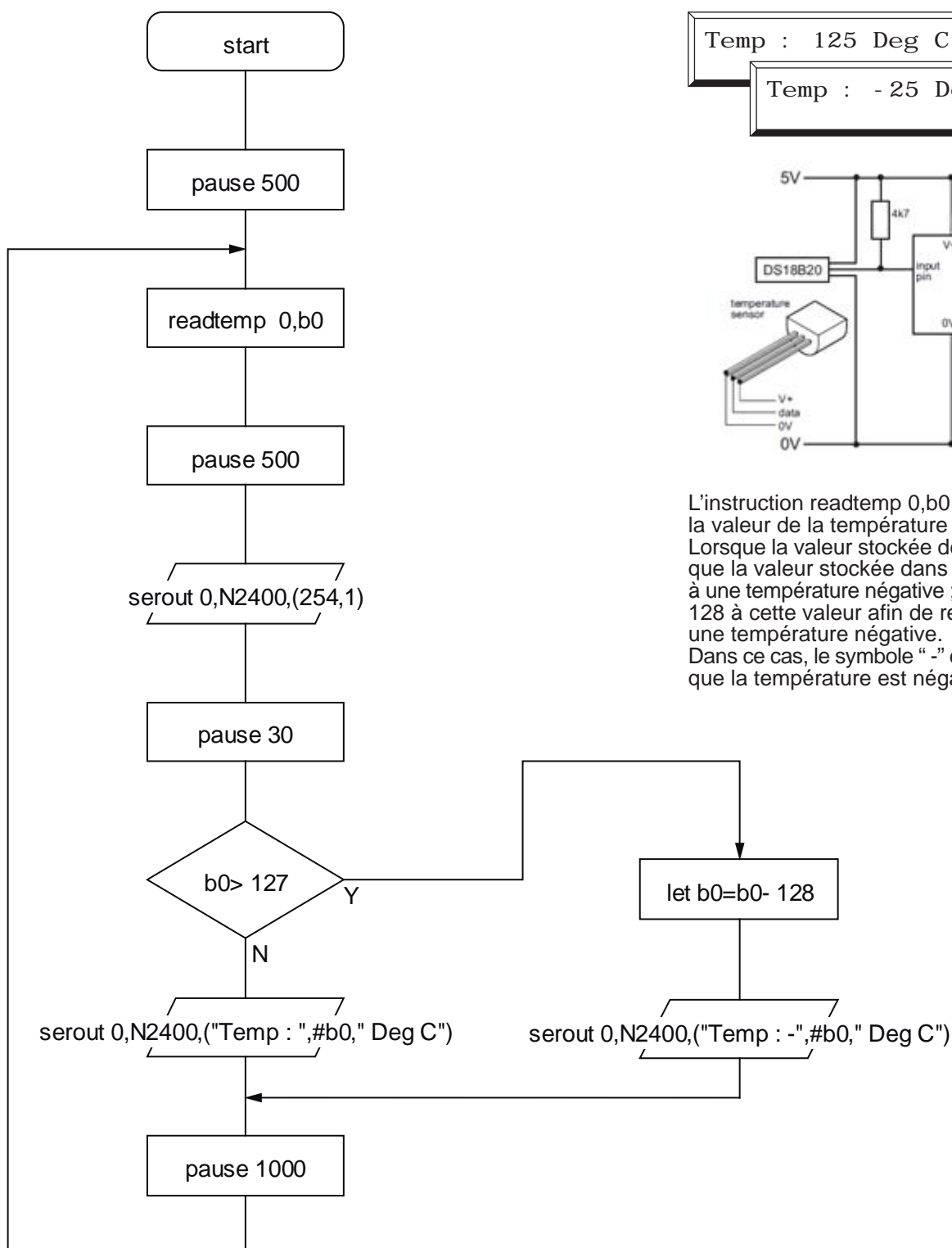


Une instruction de test vérifie cycliquement l'état du capteur à contact. Si le contact est ouvert (absence de tension sur l'entrée In 0) le message "Contact OUVERT" est affiché, sinon le message "Contact FERME" est affiché.

Acquisition et affichage de la température avec le capteur numérique DS18B20 : Programme 6-Affichage température avec capteur DS18B20.cad

Un capteur de température type DS18B20 est connecté sur l'entrée numérique In0 d'un microcontrôleur Picaxe. Ce capteur est calibré en usine. Il permet d'acquérir la valeur de la température et de la stocker directement dans la variable b0 à l'aide de l'instruction readtemp 0, b0.

Ce capteur acquiert la température sur une plage allant de -55 à + 127°C. La valeur de la température est stockée sur 8 bits dans la variable b0. Une valeur de b0 au delà de 127 correspond à une température négative ; il convient de retrancher 128 afin de calculer la température négative correspondante.

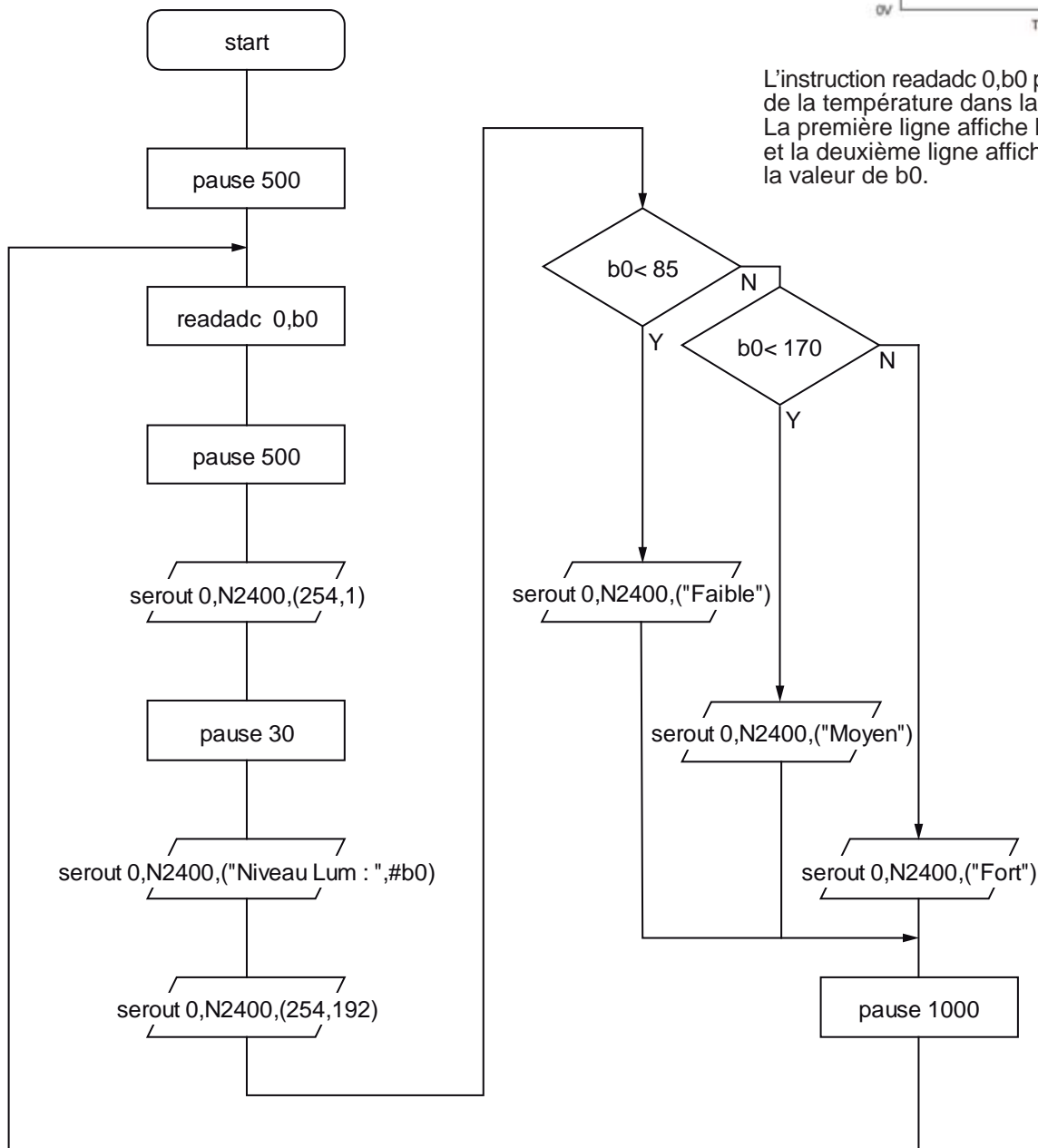
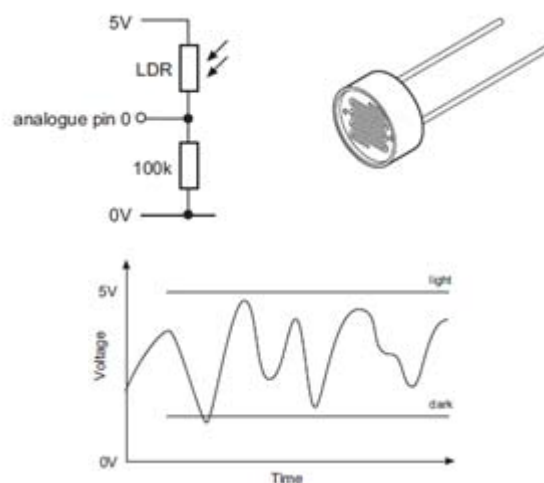
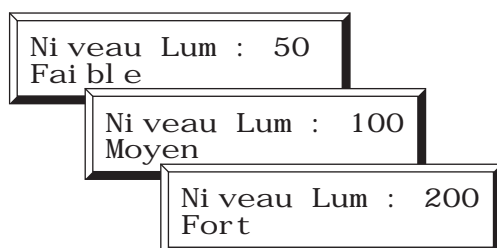


L'instruction readtemp 0,b0 permet de stocker la valeur de la température dans la variable b0. Lorsque la valeur stockée dépasse 127, cela signifie que la valeur stockée dans b0 correspond à une température négative ; il convient de retrancher 128 à cette valeur afin de retranscrire une température négative. Dans ce cas, le symbole " - " est introduit pour indiquer que la température est négative.

Acquisition et affichage du niveau de lumière capteur analogique type LDR.

Programme : 7-Affichage niveau de lumière avec capteur LDR.cad

Un capteur de lumière type LDR est connecté sur l'entrée analogique A0 d'un microcontrôleur Picaxe. Ce capteur résistif voit sa valeur varier en fonction du niveau de lumière et la tension sur l'entrée analogique A0 varie proportionnellement avec le niveau de lumière. L'instruction readadc 0, b0 permet d'exploiter cette valeur qui est convertie en une valeur numérique sur 8 bits (0 à 255) et stocké dans la variable b0. Ce capteur n'est pas calibré, il permet de mesurer des variations du niveau de lumière.



L'instruction readadc 0,b0 permet de stocker la valeur de la température dans la variable b0.

La première ligne affiche la valeur de la variable b0 et la deuxième ligne affiche un texte qui varie selon la valeur de b0.

Note : la variable b1 en début de programme permet d'initialiser la capacité du parking.



Affichage de caractères spéciaux

Le tableau suivant donne la liste des caractères gérés par l'afficheur (type Elec & Eltek).

A chaque caractère correspond une combinaison de 2 codes hexadécimaux qui permettent d'afficher le caractère souhaité. La syntaxe pour afficher un caractère spécial est :

`serout 0,N2400,($Poids FORTPoids FAIBLE)`

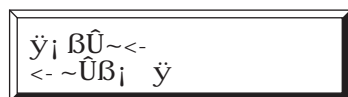
où 0 correspond à la sortie Out0 sur laquelle est connecté l'afficheur. Le code du caractère est décomposé en 2 parties (Poids FORT et Poids FAIBLE), le symbole Dollar (\$) précède le code du caractère à afficher.

| | | Poids FORT | | | | | | | | | | | | | | | |
|--------------|---|------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|
| | | 0 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | |
| Poids FAIBLE | 0 | Spé | | | | | | | | | | | | | | | |
| | 1 | Spé | | | | | | | | | | | | | | | |
| | 2 | Spé | | | | | | | | | | | | | | | |
| | 3 | Spé | | | | | | | | | | | | | | | |
| | 4 | Spé | | | | | | | | | | | | | | | |
| | 5 | Spé | | | | | | | | | | | | | | | |
| | 6 | Spé | | | | | | | | | | | | | | | |
| | 7 | Spé | | | | | | | | | | | | | | | |
| | 8 | Spé | | | | | | | | | | | | | | | |
| | 9 | Spé | | | | | | | | | | | | | | | |
| | A | Spé | | | | | | | | | | | | | | | |
| | B | Spé | | | | | | | | | | | | | | | |
| | C | Spé | | | | | | | | | | | | | | | |
| | D | Spé | | | | | | | | | | | | | | | |
| | E | Spé | | | | | | | | | | | | | | | |
| | F | Spé | | | | | | | | | | | | | | | |

Exemple : `serout 0,N2400,($DF)` affichera le caractère  

NOTE : les codes \$00 à \$0F sont prévus pour programmer des messages spéciaux prédéfinis par l'utilisateur (voir la documentation technique du module afficheur pour plus de détails).

Sous programme et affichage de caractères spéciaux. Programme : 9-Sous progs + Caractères spéciaux.cad



Ce programme affiche une sélection de caractères spéciaux.

Le programme principal appelle des sous programmes qui gèrent les éléments à afficher.

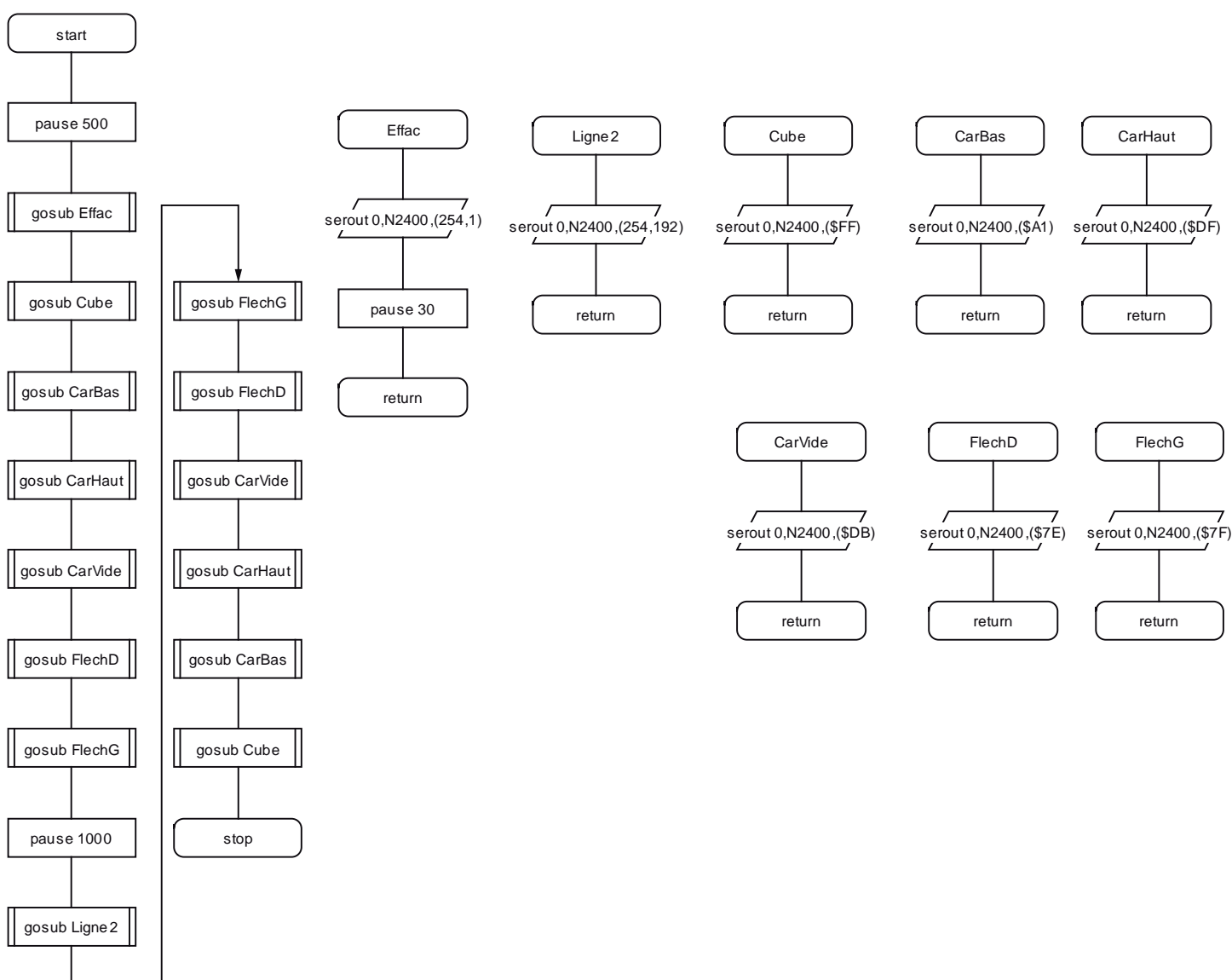
Le sous programme "Effac" efface l'affichage.

Le sous programme "Ligne2" fait un saut à la 2^{ème} ligne.

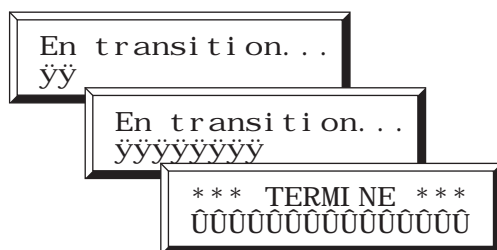
Les autres sous programmes affichent des caractères spéciaux :

- ÿ code \$FF (sous programme intitulé Cube)
- ï code \$A1 (sous programme intitulé CarBas)
- ß code \$DF (sous programme intitulé CarHaut)
- Û code \$DB (sous programme intitulé CarVide)
- ~ code \$7E (sous programme intitulé FlechD)
- <- code \$7F (sous programme intitulé FlechG)

Noter que l'emploi de sous programmes peut être utile pour combiner la gestion d'un automatisme avec l'affichage de message sur l'afficheur à cristaux liquides.



Affichage d'une barre de progression. Programme : 10-Barre de progression.cad



Affichage de En transition...sur la première ligne, progression du symbole
 ÿ répété 16 fois sur la 2^{ème} ligne,
 après 2 passages de la barre de progression, affichage de :

*** TERMI NE ***
UUUUUUUUUUUUUUUUUU

Noter que ce type de séquence peut être adapté et intégré dans un programme gérant un automatisme.

Par exemple, un bouton poussoir déclenche le départ d'une cabine d'ascenseur, le programme exécute cette séquence d'affichage de la barre de progression tant que le capteur fin de course indiquant l'arrivée de la cabine n'est pas activé, puis le programme affiche un message de service comme

CABI NE ARRI VEE

par exemple.

