

Crouzet Millenium & pilotage d'une installation solaire

Partie 1

Les fonctions de base du pilotage solaire

Un automate programmable comme le Millenium peut piloter des dispositifs complexes. On peut notamment lui faire piloter une installation solaire.

Par rapport à une régulation du commerce "toute faite", un automate programmable apporte une flexibilité qui permet d'adapter le pilotage à des installations très spécifiques. L'automate ne sera pas saturé avec votre seule installation solaire (mais cela dépend de sa capacité) : vous pourrez lui faire piloter d'autres dispositifs (le chauffage, l'installation de récupération d'eau de pluie, etc.)

Dans les rubriques qui suivent, nous allons détailler les fonctions fondamentales d'une régulation solaire et comment on peut les programmer dans un Millenium.

Les programmes donnés pour chaque fonction ne sont pas l'unique manière de la programmer mais indiquent un moyen simple. Les programmes de ces différentes fonctions peuvent être combinés pour obtenir le pilotage total de votre installation.

Ce sont en quelque sorte des briques que l'on peut combiner pour obtenir un programme complet contenant toutes les fonctions nécessaires.

Le gros avantage du Millenium est que l'on peut utiliser le logiciel de programmation Crouzet (qui tourne sur PC) sans avoir d'automate : un puissant mode de simulation est disponible et un programme peut être testé sur le PC, sans automate et sans l'environnement qu'il pilote.

Cela permet de progresser dans la programmation et de vérifier que l'on arrivera à écrire le programme souhaité avant d'acquérir l'automate.

Les seuils de température, les durées de temporisation et différents paramètres sont donnés à titre d'exemple et ils devront être ajustés sur chaque installation dans le cadre du réglage initial.

Cette première partie explique les fonctions de base d'une régulation solaire avec le Millenium.

Les parties suivantes détailleront des fonctions plus complexes, comme celles dédiées au chauffage solaire, et des fonctions utiles mais d'un usage moins courant.

Nous détaillerons les fonctions suivantes :

- La boucle solaire
- La limitation de température
- L'affichage des températures
- Les maxima des températures
- Le cumul du temps de fonctionnement
- La modification conviviale des paramètres
- Le pilotage complet d'un CESI

A noter : Bien que les programmes donnés ci-dessous aient été conçus et testés sur Millenium M3, ils doivent pouvoir s'exécuter sur Millenium M2 avec peu ou pas de modifications.

La boucle solaire

C'est la fonction indispensable à toutes les installations puisqu'elle pilote le circulateur de la boucle solaire. Elle utilise la température en haut des capteurs solaires et celle du ballon de stockage pour décider de mettre en marche ou d'arrêter le circulateur et donc la collecte de chaleur solaire.

Le Millenium a donc deux entrées analogiques reliées à deux sondes de température mesurant l'une la température en haut des capteurs solaires, l'autre la température en bas du ballon solaire à l'endroit de l'échangeur solaire.

Le circulateur est commandé par une des sorties relais.

On veut que le Millenium agisse comme suit :

- Si le capteur solaire est plus chaud que le ballon solaire, mettre en route le circulateur solaire
- Si le capteur solaire est moins chaud que le ballon solaire, arrêter le circulateur solaire

Il suffirait donc d'une simple comparaison entre les deux températures pour commander le circulateur solaire.

En pratique, il y a des points importants à prendre en compte :

1. Quand les deux températures sont très proches, de petites variations de températures peuvent faire osciller entre les deux cas, et conduire le Millenium à mettre en marche puis arrêter puis remettre en marche, etc. le circulateur rapidement sur une courte période.
2. D'autre part, la mise en marche ou arrêt du circulateur influe sur la température des capteurs solaires :
 - Quand on met en marche le circulateur solaire, on va prendre de la chaleur aux capteurs solaires, ce qui va les refroidir jusqu'à une température d'équilibre plus basse, ce qui peut suffire à les faire passer sous le seuil de fonctionnement du circulateur.
 - Quand on arrête le circulateur solaire parce que les capteurs solaire ne sont plus assez chauds, l'arrêt de prise de chaleur va les faire monter en température, ce qui peut suffire à les faire passer au dessus du seuil de fonctionnement du circulateur.
3. Pour qu'il y ait transfert de chaleur dans le serpentin du ballon solaire, il faut que la boucle solaire soit plus chaude de quelques degrés que le contenu du ballon solaire.
Il ne faut donc pas comparer la température des capteurs à celle du bas du ballon solaire, mais à une température supérieure de quelques degrés (minimum 5).

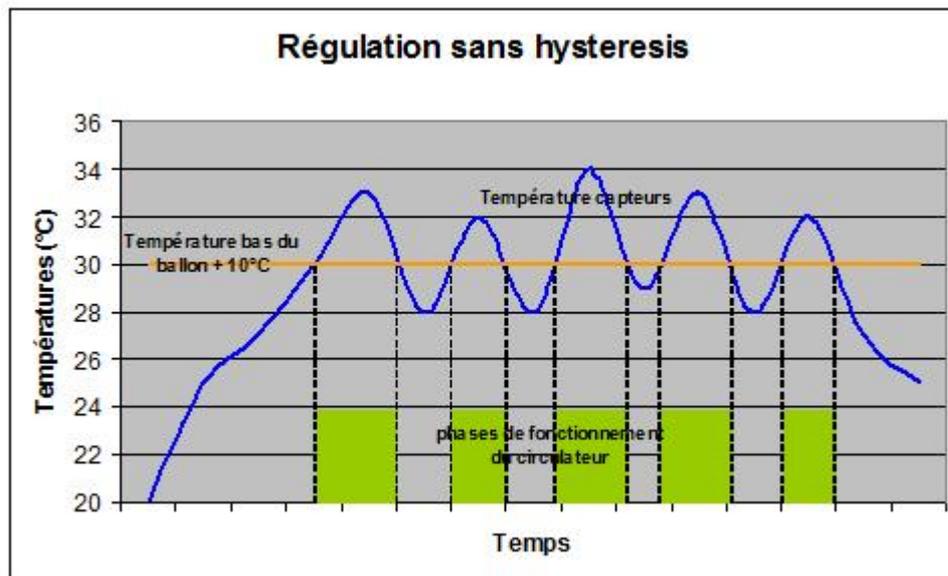
Si une seule température est utilisée pour à la fois mettre en route et arrêter le circulateur, celui-ci va osciller entre arrêt et marche.

En effet, le caloporteur est immobile dans les capteurs, il chauffe donc facilement.

Dès que le caloporteur commence à circuler, les capteurs reçoivent du caloporteur froid en entrée et donc leur température baisse. Ce qui peut conduire à passer sous la température du bas du ballon et donc arrêter le circulateur.

L'arrêt du circulateur fait que les capteurs chauffent alors un liquide immobile et le font monter facilement en température, ce qui va faire redémarrer le circulateur.

Le phénomène est illustré dans le graphique ci-dessous :



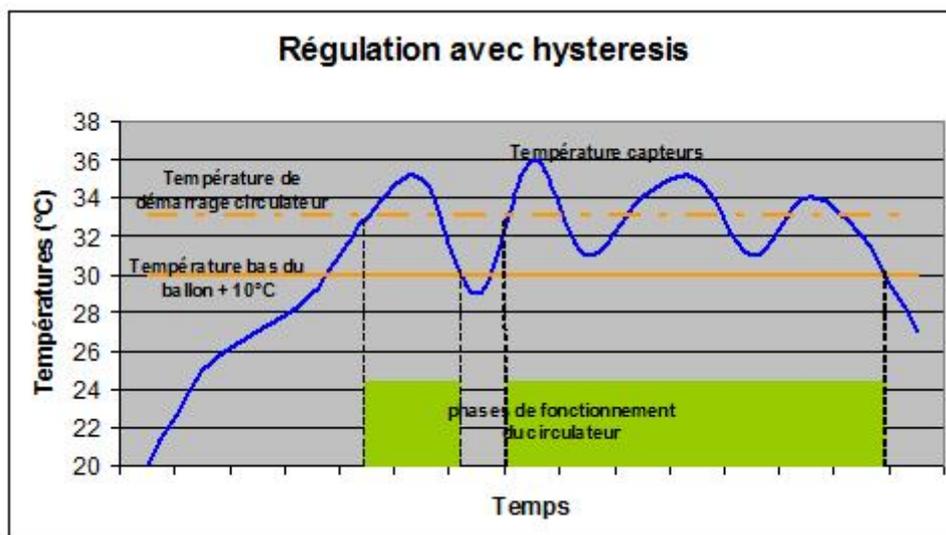
Pour éviter ce problème, on utilise deux seuils :

- un seuil de mise en route du circulateur, le plus haut des deux, typiquement égal à la température du bas du ballon solaire plus environ 10°C
- un seuil d'arrêt du circulateur, égal à la température du bas du ballon solaire plus environ 5°C.

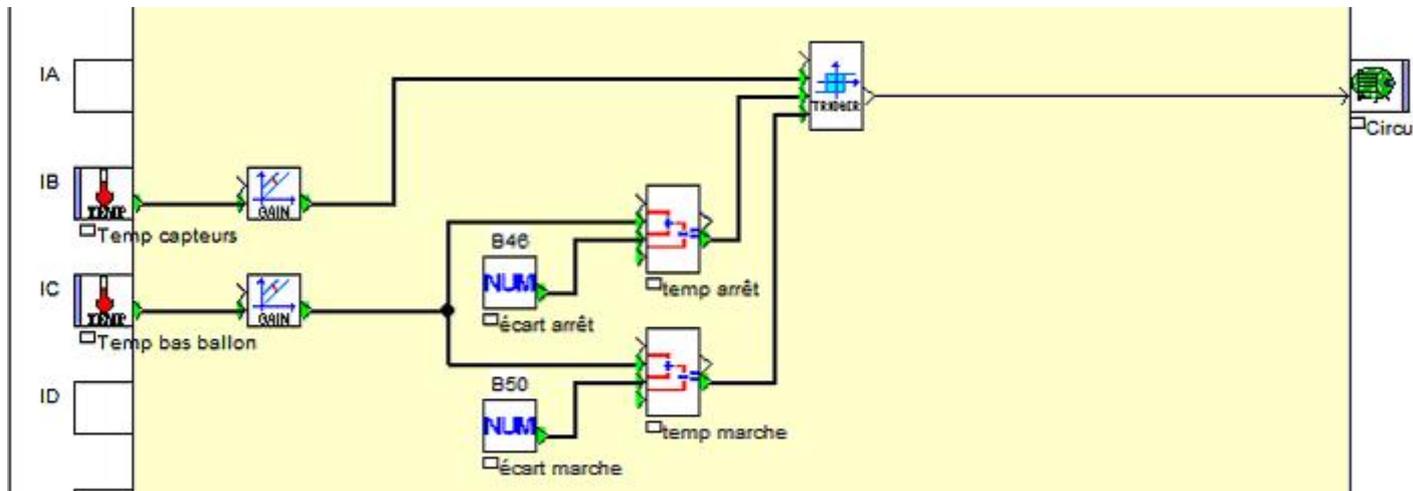
(les deux écarts sont des valeurs indicatives et devront être ajustés)

Techniquement, on dit qu'on introduit de l'hystérésis dans la commande du circulateur.

Le graphique ci-dessous illustre l'impact positif de l'hystérésis sur la commande du circulateur :



Le programme est le suivant :



[Télécharger ce programme](#)

Quelques détails sur les blocs à paramétrer

La fonction des blocs GAIN dans le programme est de convertir les valeurs données par les sondes en nombre plus significatif.

En effet, une entrée analogique du Millenium convertit une tension de 0 à 10 V en entier de 0 à 1023. C'est cet entier qui sera en sortie de chacun des deux blocs de température.

Mais à quelle température réelle correspond ce nombre entier ?

Et bien, le Millenium ne le sait pas. Il dépend des caractéristiques de la sonde de température et de l'adaptateur qui fait qu'une température est convertie en tension de 0 à 10 V.

Seul vous le savez : cela fait partie des caractéristiques de la sonde et de l'adaptateur que vous avez achetés.

On va supposer que vos sondes de température sont des sondes dont l'intervalle de température est de -10°C à $+150^{\circ}\text{C}$. La tension aux bornes de l'adaptateur de la sonde est de 0 V pour -10°C et 10 V pour $+150^{\circ}\text{C}$. Et la valeur donnée par l'entrée analogique du Millenium sera donc 0 pour -10°C et 1023 pour $+150^{\circ}\text{C}$.

Dans le Millenium et sur son affichage, vous n'allez pas recalculer en permanence à quelle température correspond ce nombre entre 0 et 1023. C'est beaucoup plus facile si le Millenium travaille directement avec les températures réelles.

C'est exactement ce que font les blocs GAIN dans ce programme : ils convertissent la valeur reçue de l'entrée analogique en un entier égal à la température mesurée par la sonde. Dans le programme, vous allez donc pouvoir ignorer qu'il y a une conversion de la température en tension (dans la sonde), de la tension en nombre de 0 à 1023 (dans le convertisseur analogique/numérique du Millenium), et de ce nombre en une valeur équivalente à la température mesurée (dans le bloc GAIN).

Dans un bloc GAIN, on définit trois constantes A, B et C. Un bloc GAIN prend son entrée, la multiplie par A, puis divise le résultat obtenu par B puis ajoute C au résultat obtenu.

Il faut donc définir les valeurs A, B et C pour nos sondes afin que la sortie du bloc GAIN soit l'entier correspondant à la température mesurée par la sonde.

Comment calculer A, B et C ?

A est l'intervalle des températures que couvre la sonde. Dans notre cas, la sonde couvre un écart de 160 degrés (150-(-10)). Donc $A = 160$.

B est l'intervalle des valeurs d'entrée, ce sera toujours 1023. Donc $B = 1023$.

C est la température quand l'entrée vaut 0. Dans notre cas, c'est -10, donc $C = -10$.

Testons sur quelques valeurs pour vérifier les valeurs trouvées.

Quand la température est -10°C , nous avons vu que l'entrée analogique donne 0. La sortie du bloc GAIN sera alors $0 \times 160 / 1023 - 10 = -10$. C'est correct.

Quand la température est $+150^{\circ}\text{C}$, nous avons vu que l'entrée analogique donne 1023. La sortie du bloc GAIN sera alors $1023 \times 160 / 1023 - 10 = 150$. C'est correct.

Quand la température est au milieu de la gamme de mesure de la sonde (70°C), l'entrée analogique va donner la moitié des valeurs possibles, soit 512. La sortie du bloc GAIN sera alors $512 \times 160 / 1023 - 10 = 70$. C'est encore correct.

Ça marche, nous pouvons entrer les valeurs A, B et C dans les blocs GAIN de notre programme. Nous supposons ici que les deux sondes sont identiques. Si ce n'est pas le cas, le calcul des valeurs A, B et C devra être fait pour chaque modèle de sonde.

Pour les blocs NUM, le paramétrage est plus simple, puisqu'il suffit de spécifier quel écart en température nous souhaitons :

- Pour l'écart définissant le seuil d'arrêt, nous pouvons mettre 5, ce qui signifie que le Millenium arrêtera le circulateur quand la température des capteurs descendra sous la température du bas du ballon plus 5°C .
- Pour l'écart définissant le seuil de mise en marche, nous pouvons mettre 10, ce qui signifie que le Millenium mettra en marche le circulateur quand la température des capteurs montera au dessus de la température du bas du ballon plus 10°C .

Nous avons déjà mentionné que les écarts des seuils marche et arrêt pourraient être ajustés sur l'installation. Nous allons dans un premier temps les rendre modifiables via la face avant du Millenium. Ainsi leur modification ne nécessitera pas le téléchargement d'un nouveau programme.

Pour cela, dans la boîte de dialogue NUM, il faut cocher "Modification autorisée".

A noter que pouvoir modifier les paramètres des blocs GAIN n'a pas d'intérêt, on ne permet donc pas leur modification à partir de la face avant.

Il faut également noter les numéros de blocs modifiables et leur signification.

Lorsque le programme s'exécute sur le Millenium, à partir de l'affichage des entrées et sorties, il faut procéder comme suit :

1. Presser OK puis ESC en même temps, le Millenium affiche trois lignes ARRET, PARAMETRES, DIVERS.
2. Sélectionner PARAMETRES avec les touches + et - puis valider par OK.

3. Le Millenium fait choisir le bloc à modifier parmi tous les blocs modifiables du programme. Faire défiler les numéros de bloc avec les touches + et - et sélectionner avec OK.
4. Pour un bloc NUM, le Millenium indique VALEUR CONSTANTE et permet de la modifier avec + et -, puis de valider par OK.

Dès la modification faite, le programme du Millenium utilise la nouvelle valeur.

A noter : il existe heureusement une méthode moins rustique et moins source d'erreurs pour changer la valeur des paramètres. Elle est décrite plus loin.

La limitation de température

Le programme de la section précédente va laisser en marche le circulateur tant que de la chaleur solaire peut être captée et transférée au ballon solaire.

Il est fortement recommandé d'éviter de porter un stock d'eau solaire à trop haute température, notamment pour un ballon ECS solaire si l'eau est calcaire (la chaleur solaire peut même faire bouillir l'eau d'un ballon). Cette section va expliquer comment limiter le chauffage du stockage solaire à une certaine température.

Nous allons définir une consigne de température maximale pour notre ballon solaire. Lorsque cette température sera atteinte, le circulateur s'arrêtera et ne se mettra plus en marche, tant que le bas du ballon ne descendra pas sous cette consigne.

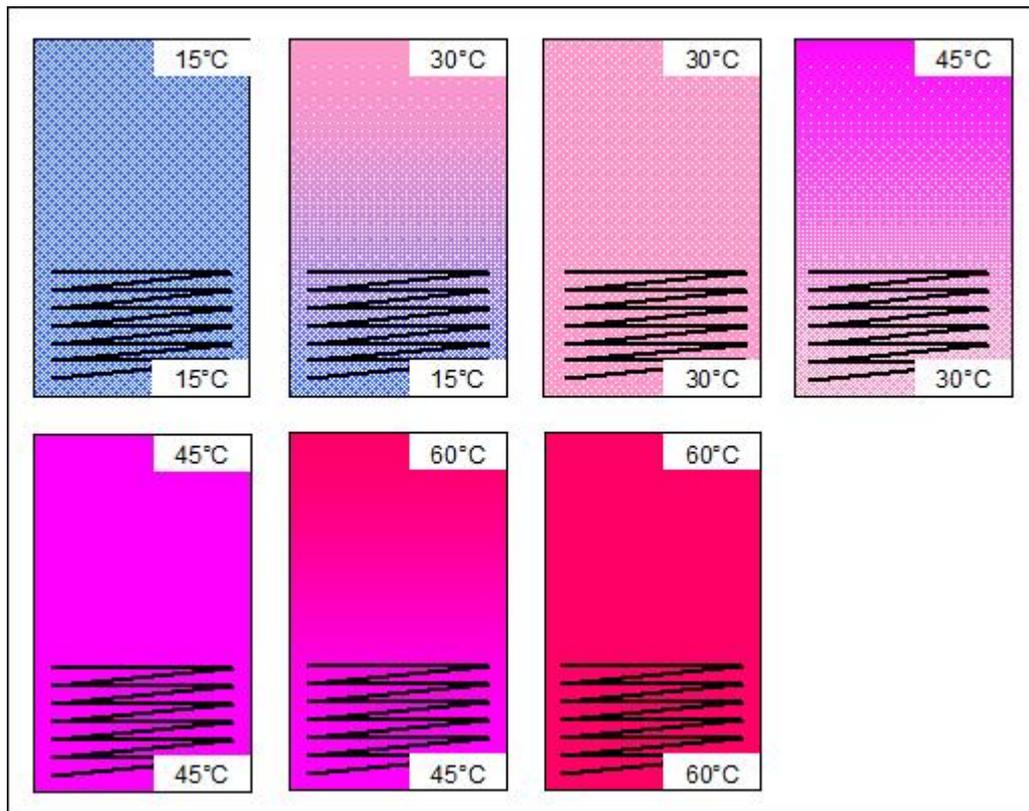
Nous n'avons pas de sonde de température en haut du ballon or c'est en haut que l'eau est la plus chaude (plus l'eau est chaude, plus elle est légère).

En pratique, pendant la chauffe solaire d'une installation solaire standard, le haut du ballon ne peut pas être chauffé à haute température si le bas est encore froid. En effet, la puissance solaire permet d'élever l'eau du bas du ballon de quelques degrés (par exemple 10°C).

Tant que le bas du ballon est froid, le solaire transforme cette eau froide en eau tiède (qui monte dans le ballon). Ce n'est que lorsque tout le bas du ballon est tiède (et que donc tout le ballon est tiède) que le ballon va monter en température.

Il suffit donc de limiter la température du bas du ballon pour limiter celle du haut.

Le dessin suivant schématise le chauffage progressif d'un ballon solaire, en l'absence de tirage d'eau chaude :



On voit également qu'il est important qu'un ballon solaire ne soit pas trop grand par rapport à l'énergie que les capteurs solaires peuvent collecter. Si c'est le cas, on risque d'avoir beaucoup d'eau tiède le soir, alors que l'on veut de l'eau chaude, même en moins grande quantité.

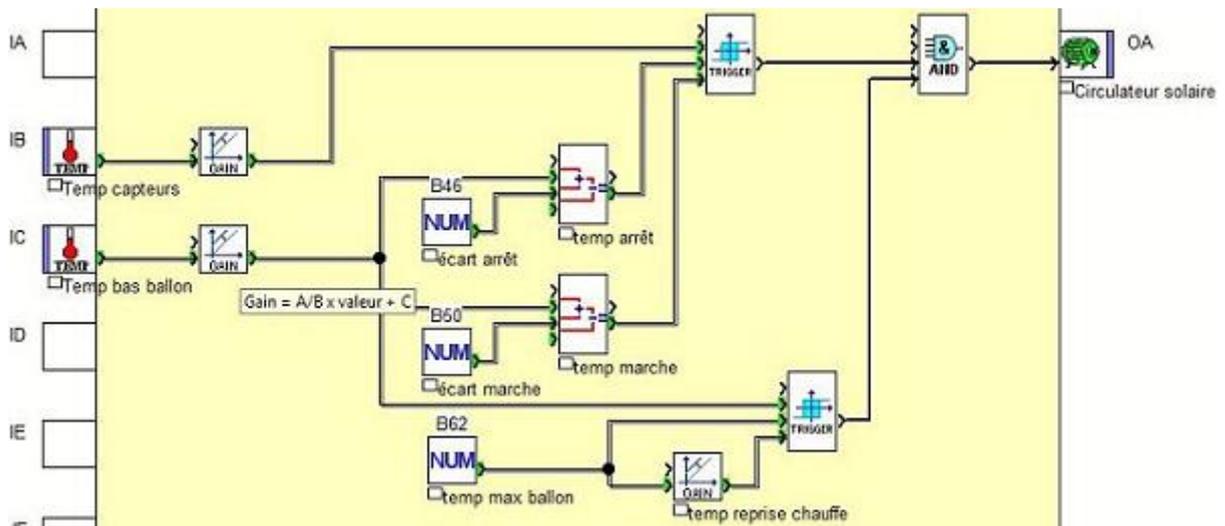
Une simple comparaison de la température du ballon avec celle de la valeur maximale pourrait suffire mais nous voulons éviter les oscillations. Nous utilisons donc un TRIGGER pour différencier les seuils d'arrêt et de marche :

- Nous interdisons au solaire de chauffer le ballon quand la sonde de bas de ballon atteint la température maximale.
- On peut à nouveau chauffer le ballon si le bas du ballon est 4°C (valeur arbitraire) sous la température maximale.

Dans le programme, nous devons donc soustraire une constante à la valeur d'un bloc NUM qui définit la température maximale. Nous pourrions utiliser un bloc +- et un autre bloc NUM pour définir la constante à soustraire.

Nous utilisons un bloc GAIN directement, ce qui permet d'utiliser un bloc de moins. On paramètre A=1, B=1 et on fixe C à l'opposé de la constante (-4 pour un écart de 4°C).

Le programme :



[Télécharger ce programme](#)

La température maximale est un paramètre que l'on voudra pouvoir modifier sur l'installation. On rend donc le bloc NUM correspondant modifiable.

Par contre, pouvoir modifier les écarts par rapport à la valeur maximale pour le bloc TRIGGER n'a pas d'intérêt. Il est donc inutile de pouvoir modifier les paramètres du bloc GAIN.

L'affichage des températures

Pour des raisons informatives ou de diagnostic (bogue dans le programme ou problèmes divers), il est important de pouvoir obtenir les températures dans l'installation telles que mesurées par l'automate programmable.

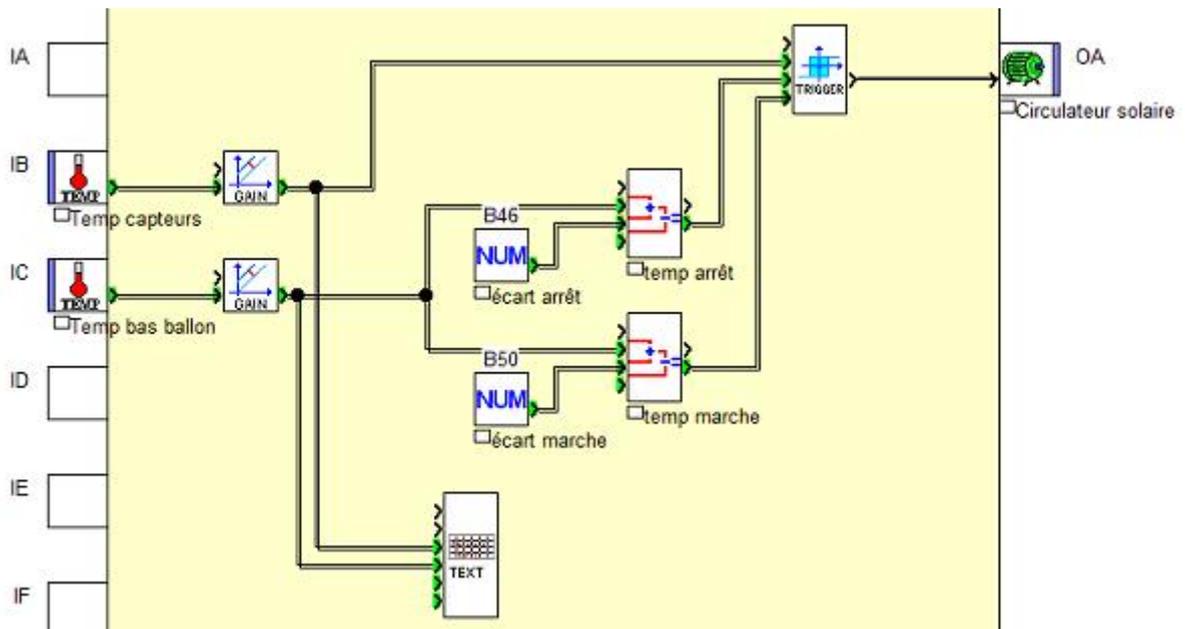
Cette fonction permet d'afficher sur l'écran du Millenium les températures actuelles mesurées par les sondes.

Nous supposons que l'installation est classique et qu'elle possède deux sondes, l'une en haut des capteurs, l'autre en bas du ballon solaire.

Nous voulons que les températures mesurées soient l'affichage par défaut du Millenium quand le programme est en fonctionnement.

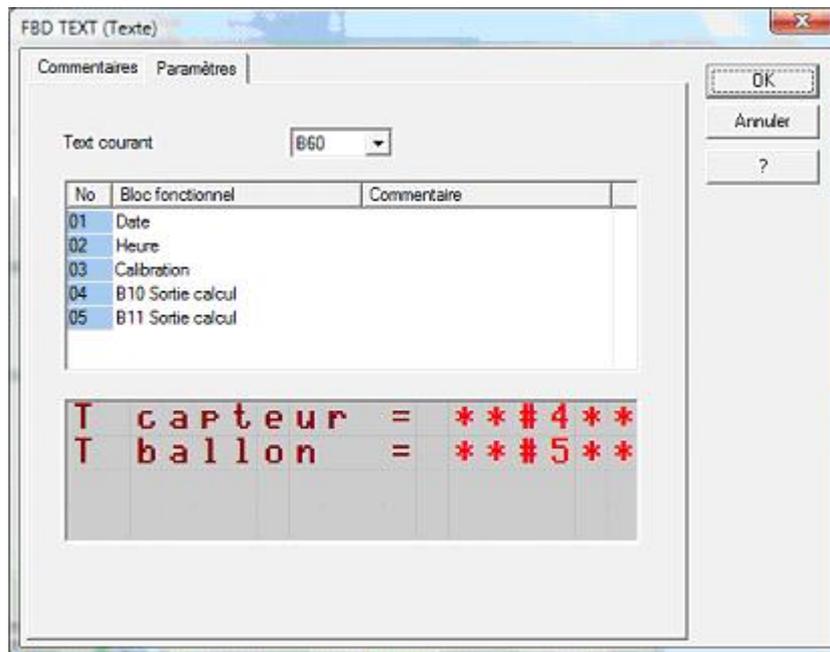
Nous utilisons un bloc TEXT qui permet de définir un écran de Millenium complet, contenant du texte et jusqu'à 4 valeurs à afficher.

Voici notre programme de boucle solaire, juste modifié pour l'affichage des températures :

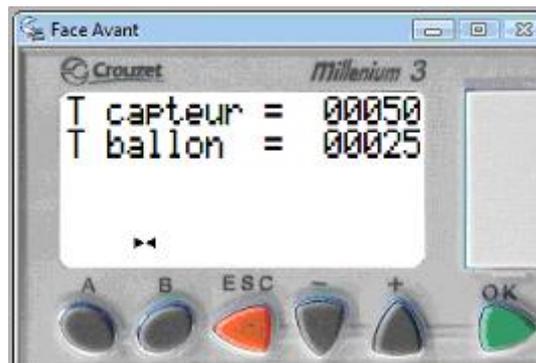


[Télécharger ce programme](#)

Le bloc TEXT est paramétré comme ci-dessous :



Et voici l'écran du Millenium quand le programme s'exécute :



A noter : Lorsque cet écran est affiché :

- On peut voir l'écran par défaut du Millenium (celui donnant l'état des entrées/sorties) en maintenant appuyée la touche ESC.
- On peut accéder au menu du Millenium en appuyant sur OK puis ESC en même temps.

Les maxima des températures

Il peut être utile d'archiver les températures extrêmes mesurées par les sondes. Cette fonction permet de gérer ces valeurs maximales, de les afficher et de les ré-initialiser à la demande.

Nous allons utiliser le bloc MIN MAX. Ce bloc mémorise les valeurs minimales et maximales de son entrée, tant qu'il n'est pas réinitialisé. Il positionne deux sorties aux valeurs minimale et maximale mémorisées.

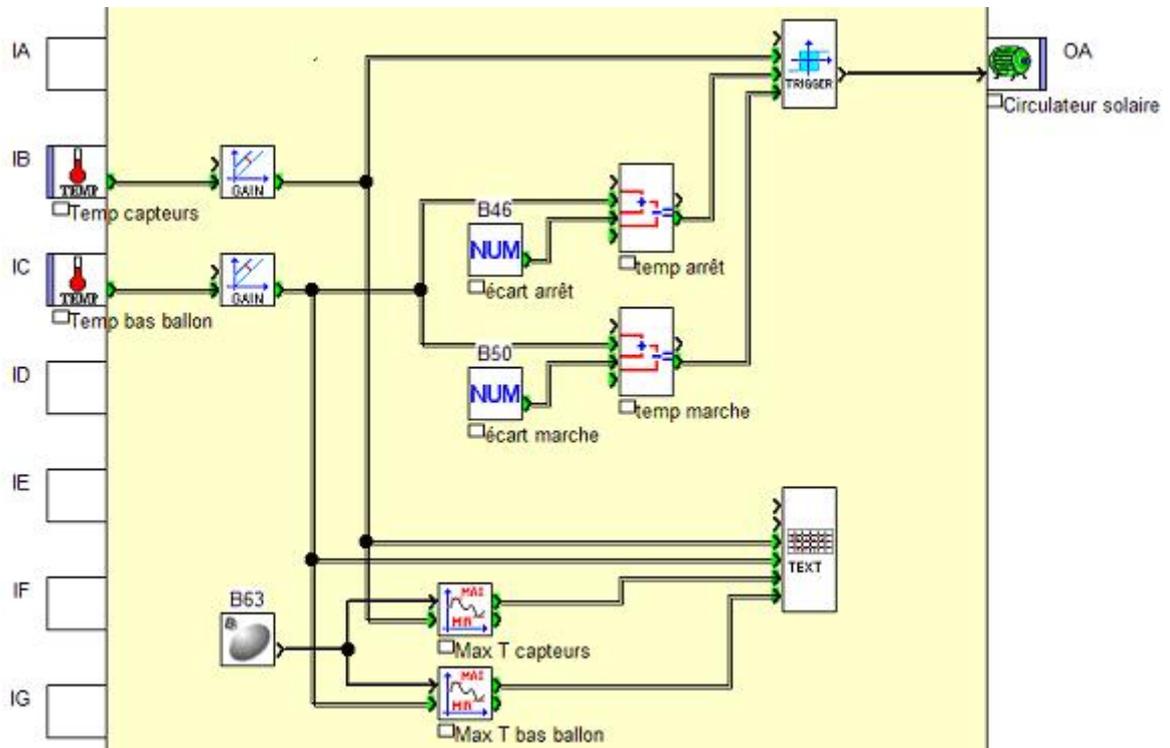
Il nous faut ajouter deux choses :

- un moyen d'afficher les valeurs maximales actuelles
- un moyen de remettre à zéro les valeurs maximales mémorisées

Pour l'affichage, nous allons utiliser l'affichage de température de la section précédente que nous allons simplement modifier pour afficher en plus nos deux valeurs maximales.

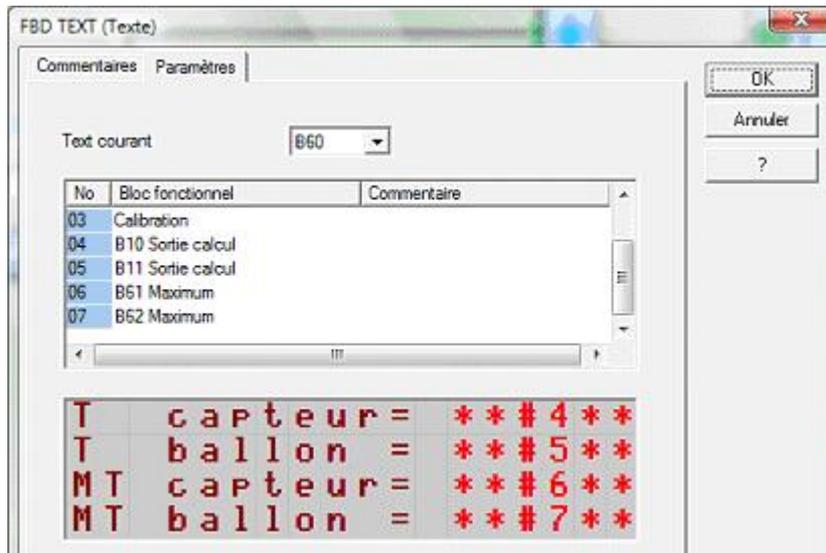
Pour la remise à zéro, nous allons utiliser le bouton B de la face avant.

Cela donne le programme suivant :

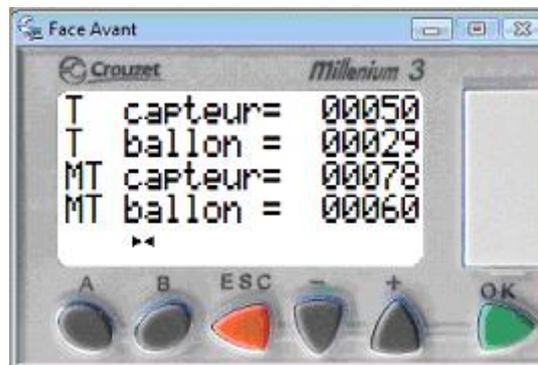


[Télécharger ce programme](#)

Avec le bloc TEXT modifié comme suit :



Quand le programme s'exécute, le Millenium affiche l'écran suivant :



A noter : nous avons coché la case "Sauvegarde sur coupure secteur" des blocs MIN MAX, ce qui permet de conserver nos valeurs maximales même en cas de coupure de l'alimentation du Millenium.

Le cumul du temps de fonctionnement

Lorsque vous êtes absent, l'installation solaire continue (heureusement) de fonctionner.

A votre retour (du travail, de week-end, de vacances), vous voulez vérifier si l'installation a fonctionné et combien de temps.

Cette fonction permet de cumuler le temps de fonctionnement du circulateur solaire, par jour et par mois.

Nous allons utiliser le bloc PRESET H-METER qui permet de cumuler le temps pendant lequel son entrée est à 1.

Ce bloc peut aussi mettre à 1 sa sortie quand un temps donné est atteint, mais nous n'utiliserons pas cette possibilité.

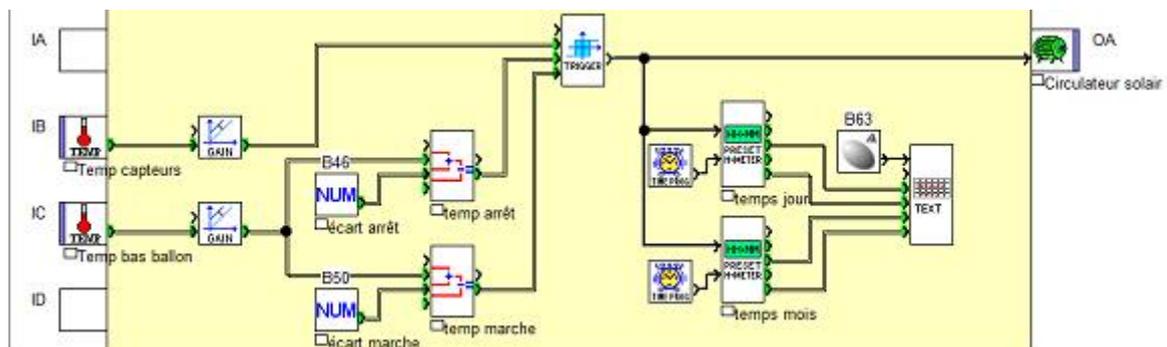
Un bloc PRESET H-METER sera utilisé pour chaque cumul : journalier et mensuel.

Pour la remise à zéro des blocs PRESET H-METER, nous allons utiliser des blocs TIME PROG qui positionneront les entrées "Remise à zéro" à 0h00 tous les jours pour le cumul journalier et à 0h00 le 1er de chaque mois pour le cumul mensuel.

Il faut pouvoir afficher ces cumuls de temps à la demande. En effet, ce ne sont pas les informations les plus importantes à afficher (l'affichage des températures des sections précédentes l'est bien plus).

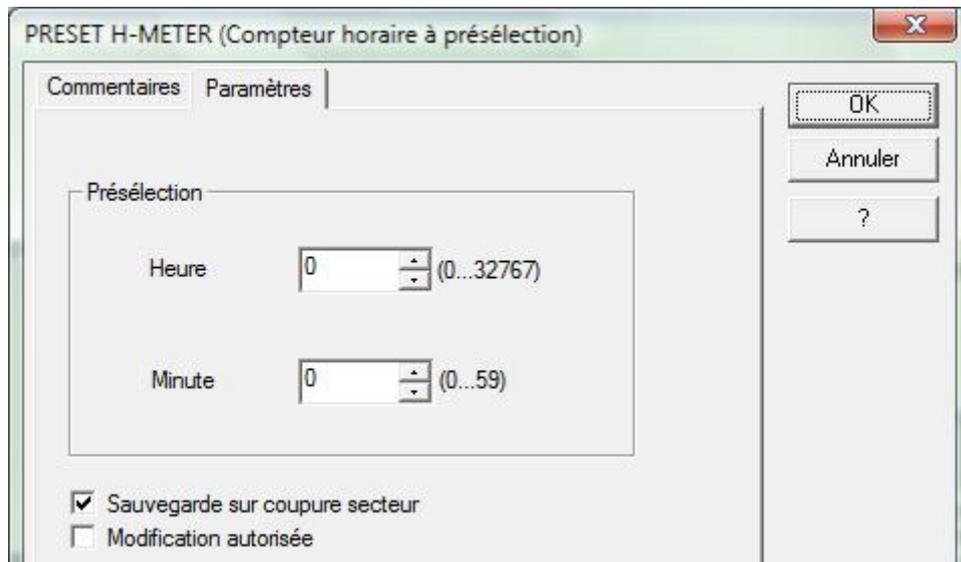
Nous allons donc utiliser un autre bloc TEXT, qui s'affichera quand on appuiera sur la touche A.

Voici le programme :

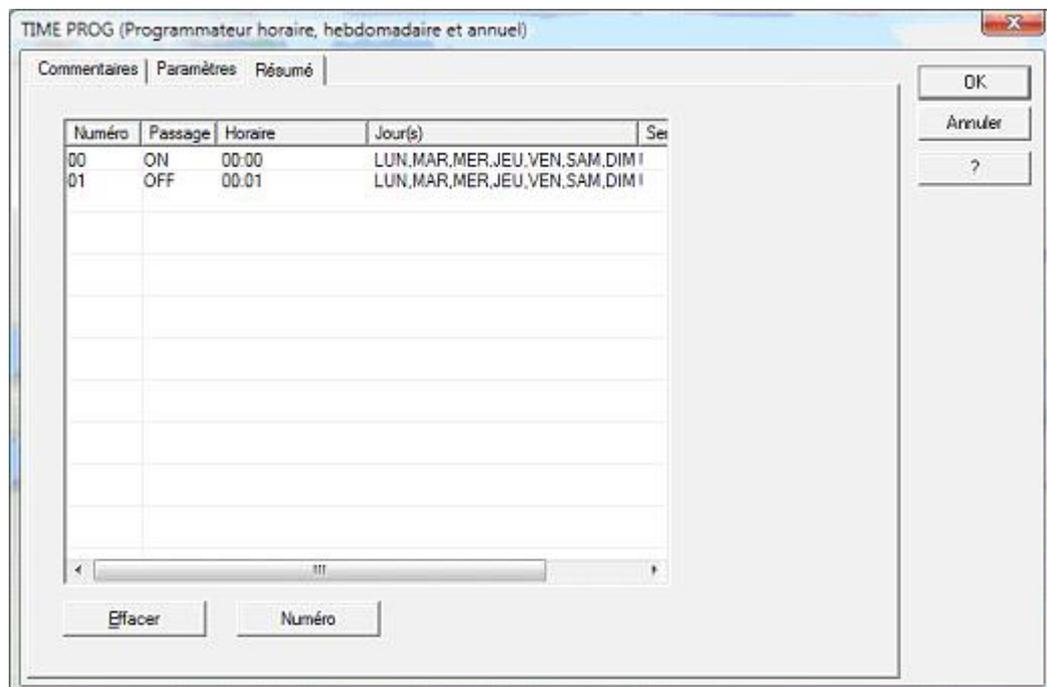


[Télécharger ce programme](#)

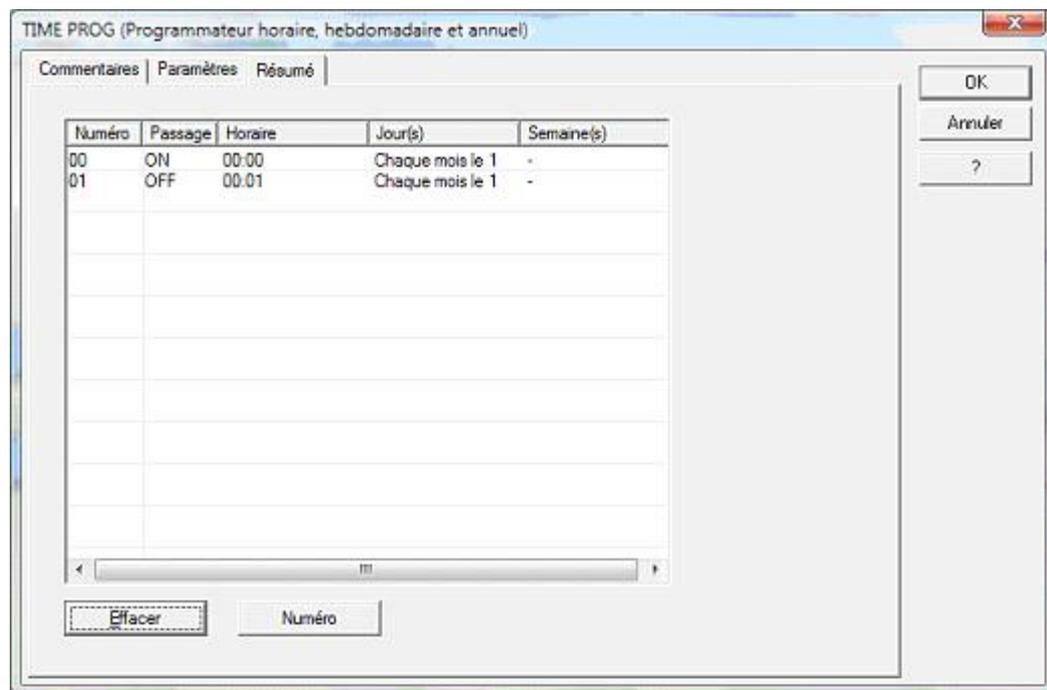
Les deux blocs PRESET H-METER sont paramétrés comme suit (notez que nous cochons la case "Sauvegarde sur coupure secteur" afin d'éviter de perdre nos temps cumulés en cas de perte d'alimentation du Millenium) :



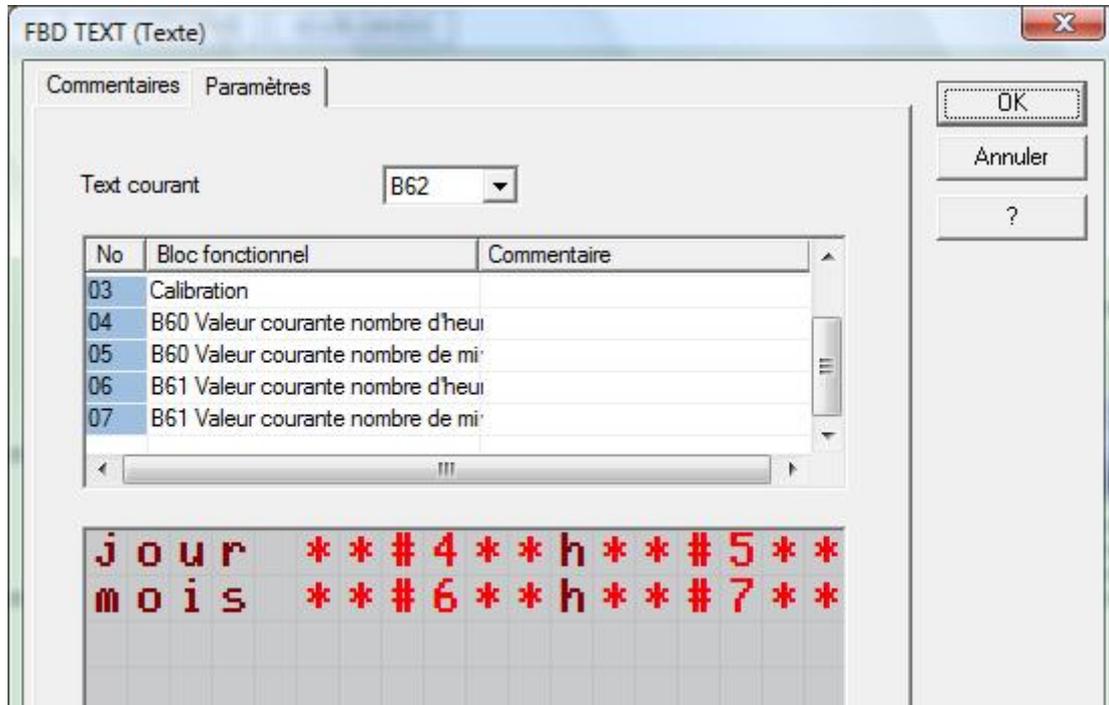
Le bloc TIME PROG chargé de la remise à zéro quotidienne est paramétré comme ci-dessous. A noter que l'on programme le plus petit intervalle pendant lequel la sortie peut passer à 1 (1 minute) La sortie du bloc sera à 1 entre 0h00 et 0h01 tous les jours.



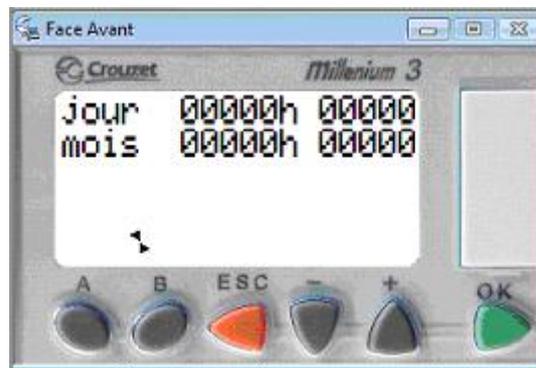
Le bloc TIME PROG chargé de la remise à zéro mensuelle est paramétré comme suit :



Enfin le paramétrage du bloc TEXT affichant les cumuls :



Quand on appuie sur le bouton A, l'écran suivant s'affiche :



L'affichage n'est pas élégant : le bloc TEXT n'autorise pas le formatage des nombres et ceux-ci sont toujours affichés sur au moins 5 caractères.

Pour réaliser un meilleur affichage, il faut utiliser des blocs DISPLAY, qui permettent de mieux définir le format des nombres. Par contre, chaque bloc DISPLAY n'affiche qu'un nombre ou une suite de caractères fixes, il en faudrait donc plusieurs pour décrire notre écran.

La modification conviviale des paramètres

Nous avons vu comment modifier les paramètres de certains blocs à partir de la face avant du Millenium, via le menu PARAMETRES. Cette manière de faire ne nécessite aucune programmation, mais elle a beaucoup d'inconvénients :

- elle est peu pratique
- elle nécessite de garder à côté du Millenium les numéros des blocs NUM à modifier et leur signification
- elle est impossible à utiliser pour un utilisateur ne connaissant pas le Millenium

En bref, c'est un moyen utilisable pendant la programmation, mais inadapté à l'utilisation du programme. Nous allons voir comment remédier à ces problèmes via un peu de programmation.

Nous allons reprendre le programme de boucle solaire avec limitation de la température du ballon. Dans ce programme, trois paramètres doivent pouvoir être modifiés après la programmation :

- l'écart de mise en marche du circulateur solaire
- l'écart d'arrêt du circulateur solaire
- la température maximale du ballon

Le Millenium permet l'affichage de valeurs via les blocs DISPLAY et TEXT. A chaque fois qu'une valeur d'entrée change, son affichage est mis à jour automatiquement.

Le Millenium permet aussi qu'une valeur affichée par ces blocs soit modifiée via la face avant pendant son affichage. Ainsi un bloc NUM affiché par un bloc DISPLAY peut avoir sa valeur modifiée par l'utilisateur.

Nous allons modifier le programme pour permettre la modification de nos 3 paramètres.

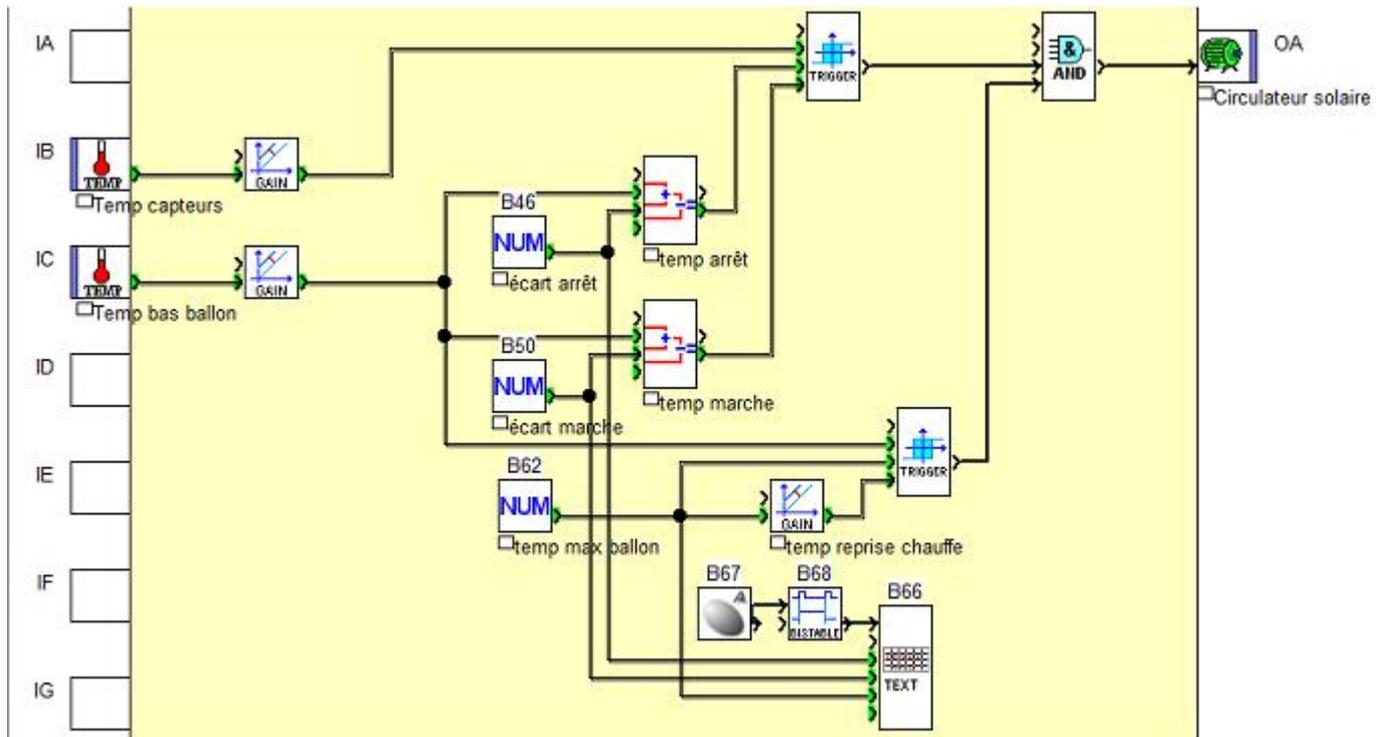
Quand on va appuyer sur le bouton A (de la face avant), le Millenium va afficher les valeurs des trois

paramètres avec leur signification. Les valeurs pourront être modifiées si nécessaire. Un autre appui sur le bouton A fera revenir l'affichage par défaut.

L'affichage et la modification sont faits par un seul bloc TEXT. La bascule entre l'écran des paramètres et l'affichage par défaut est faite par un bloc BISTABLE commandé par la touche A.

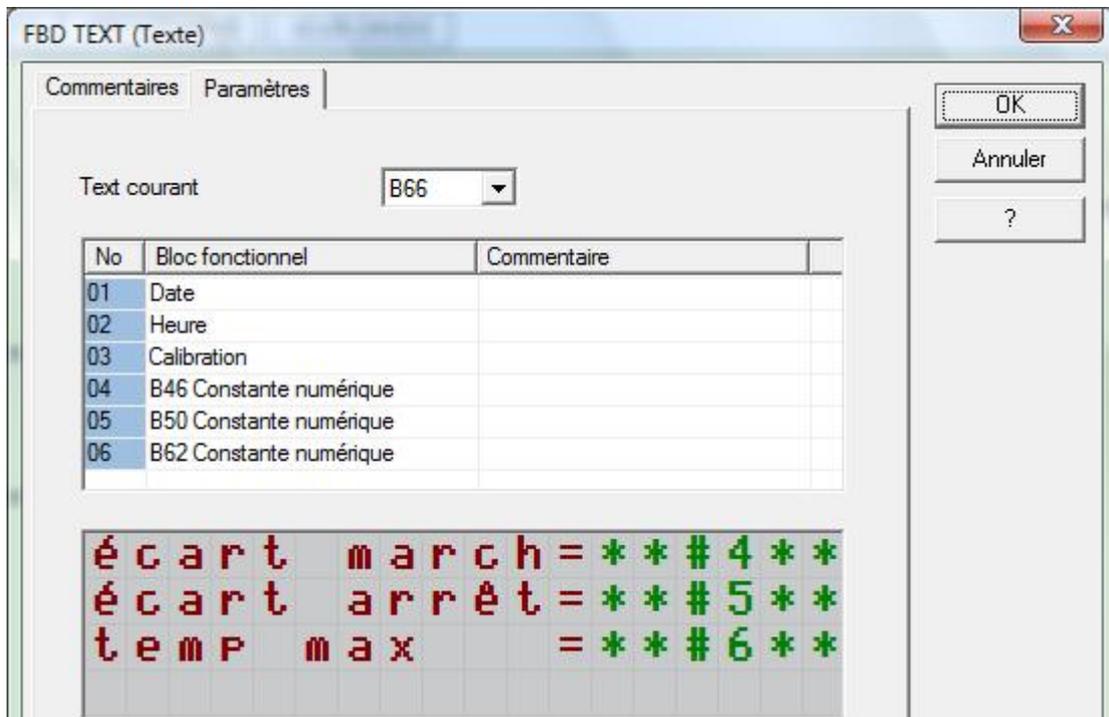
La modification ne demande donc que trois blocs de plus.

Voici le programme :



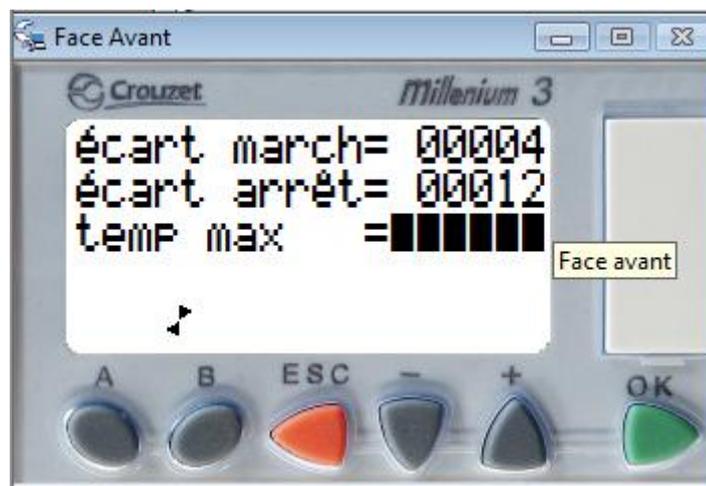
Le paramétrage du bloc TEXT est expliqué ci-dessous.

Dans la partie définition de l'écran du Millenium, une valeur non modifiable apparaît en rouge, une valeur modifiable apparaît en vert. La couleur change quand la valeur est sélectionnée et que l'on coche ou décoche l'option "Modification autorisée" de la valeur.



A noter : il n'est pas nécessaire que l'option "Modification autorisée" soit cochée dans les blocs NUM.

L'affichage quand on appuie sur le bouton A :



Pour modifier la valeur d'un champ :

- On sélectionne le paramètre que l'on veut modifier avec + et - puis OK.
- On ajuste la valeur du paramètre avec + et - puis on la valide par OK (la valeur modifiée est prise en compte immédiatement par le programme).

On peut rester sur cet écran autant de temps que l'on veut. Un appui sur A ou ESC permet de revenir à l'affichage par défaut.

Dans le cas où l'on a plus de quatre paramètres à modifier (ou plusieurs écrans d'information), on peut utiliser plusieurs blocs TEXT de ce type. Pour les activer l'un derrière l'autre, on peut utiliser un bloc CAM commandé par une touche. Chaque appui sur la touche fait passer à l'écran TEXT suivant et donc aux paramètres suivants. On peut programmer ESC pour revenir à l'écran par défaut. Ce procédé est utilisé et détaillé dans la rubrique suivante.

Le pilotage complet d'un CESI

Rien de nouveau dans cette rubrique. Simplement un programme regroupant ensemble les fonctions que nous venons de voir :

- la boucle solaire
- la limitation de température
- l'affichage des températures
- le maximum des températures
- le cumul du temps de fonctionnement
- la modification conviviale des paramètres

Ce programme constitue le pilotage complet d'un CESI (Chauffe-Eau Solaire Individuel).

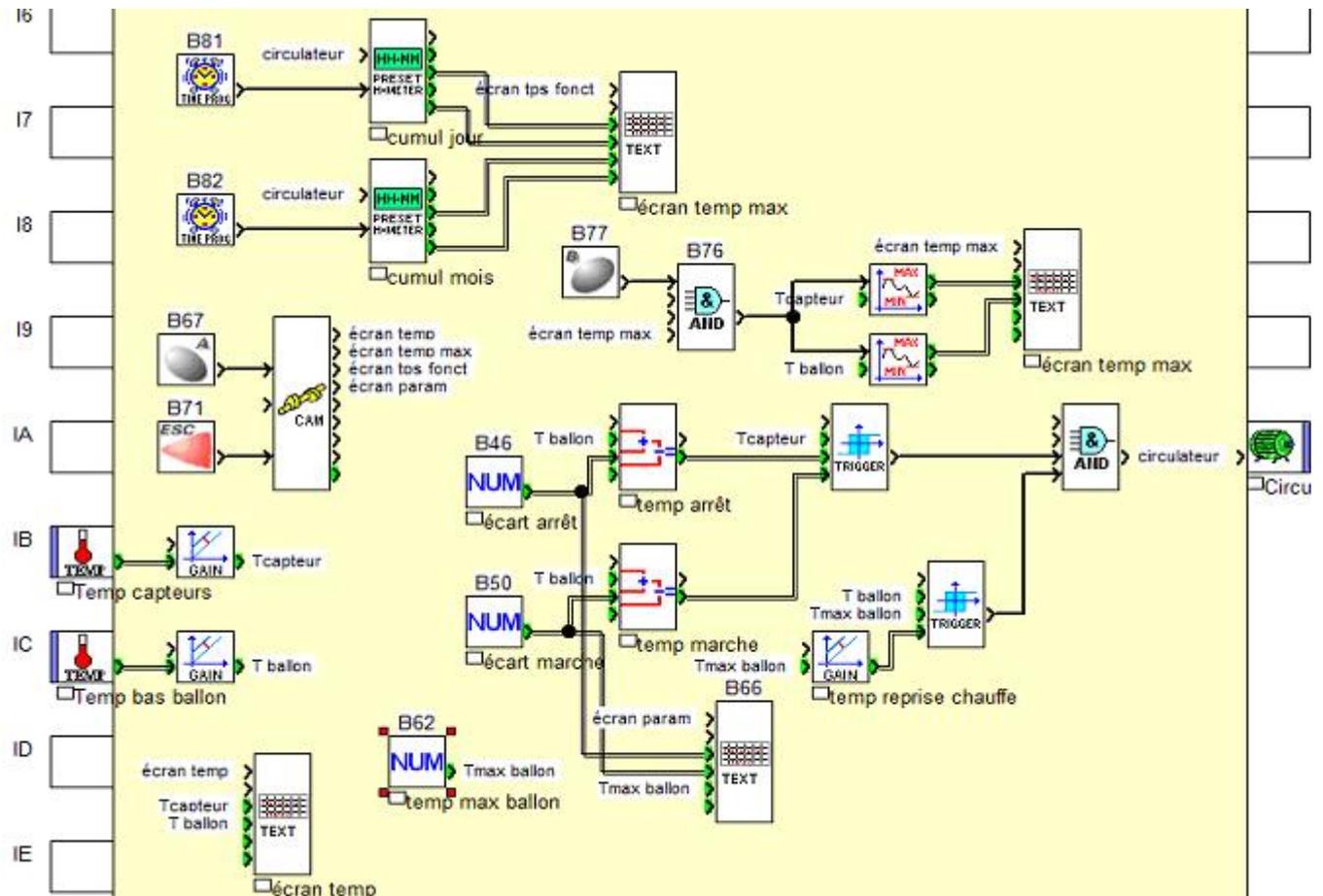
Les différents affichages et la manière d'y accéder sont regroupés pour améliorer le confort d'utilisation. En appuyant successivement sur le bouton A, on passe sur une série de quatre écrans successifs :

1. les températures actuelles (écran par défaut, pas de modification possible)
2. les températures maximales (un appui sur le bouton B sur cet écran remet les maxima à zéro)
3. le cumul journalier et mensuel des temps de fonctionnement (aucune modification possible)
4. la modification des paramètres (tous les paramètres peuvent être modifiés)

Un appui sur ESC sur un des écrans fait revenir aux températures actuelles.

Le programme a changé sur un autre aspect : certaines liaisons très utilisées ne sont plus "filaire" mais "texte". En effet, quand un programme devient un peu conséquent, les liaisons filaires forment un réseau de traits difficile à lire. Dans ce cas, il est préférable de nommer les liaisons les plus utilisées en leur donnant un nom significatif.

Le programme se transforme ainsi en petits blocs logiques, utilisant des données nommées. Avec un peu d'habitude et en choisissant des noms de liaisons courts et précis, c'est bien plus lisible qu'un réseau emmêlé de liaisons filaires.



[Télécharger ce programme](#)

Le programme utilise 26 blocs mais seulement 11 font fonctionner l'installation. Les autres gèrent des fonctions annexes, les affichages et modifications de paramètres.

Le détail de la mécanique d'affichage

Le bloc CAM du Millennium simule le comportement d'un programmeur à cames. Le bloc a un certain nombre d'états ordonnés (jusqu'à 50), dont un seul est actif à la fois. Dans chaque état, on peut définir l'état des 8 sorties (0 ou 1). Les entrées d'un bloc CAM permettent d'avancer à l'état suivant, de revenir à l'état précédent ou de retourner à l'état initial.

On se sert de ce bloc pour gérer la séquence d'écrans successifs (4 dans notre cas). Un état correspond à un écran (donc on a 4 états pour le bloc CAM). Chaque bloc TEXT est connecté à une seule sortie du bloc CAM. Dans chaque état, une seule sortie est active : celle reliée à l'entrée SET du bloc TEXT qui décrit son écran.

L'entrée pour avancer à l'état suivant est connectée au bouton A. Ainsi chaque pression sur le bouton A fait avancer le bloc CAM à l'état suivant, et active donc le bloc TEXT qui décrit l'écran suivant.

L'entrée de retour à l'état initial est connectée à la touche ESC, cela fait revenir le bloc CAM à l'état initial, donc au 1er écran.

Le Millenium n'affiche qu'un seul bloc TEXT à la fois (qui définit tout l'écran). Si plusieurs blocs TEXT sont actifs en même temps, c'est celui qui a le numéro de bloc le plus élevé qui s'affiche. Ce n'est pas le cas dans ce programme où le bloc CAM assure qu'un seul bloc TEXT est actif à la fois.

Le paramétrage du bloc CAM est détaillé ci-dessous, on voit bien les 4 états et le fait que chacun active une seule sortie du Millenium :

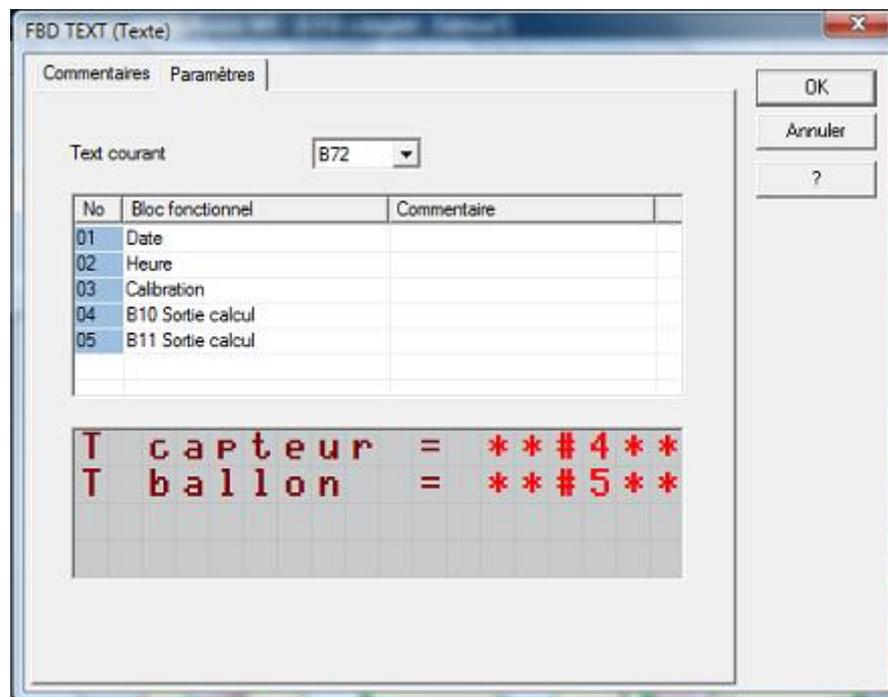
Number of steps in the program: 4

Position	S1	S2	S3	S4	S5	S6	S7	S8
0	1	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0
3	0	0	0	1	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0

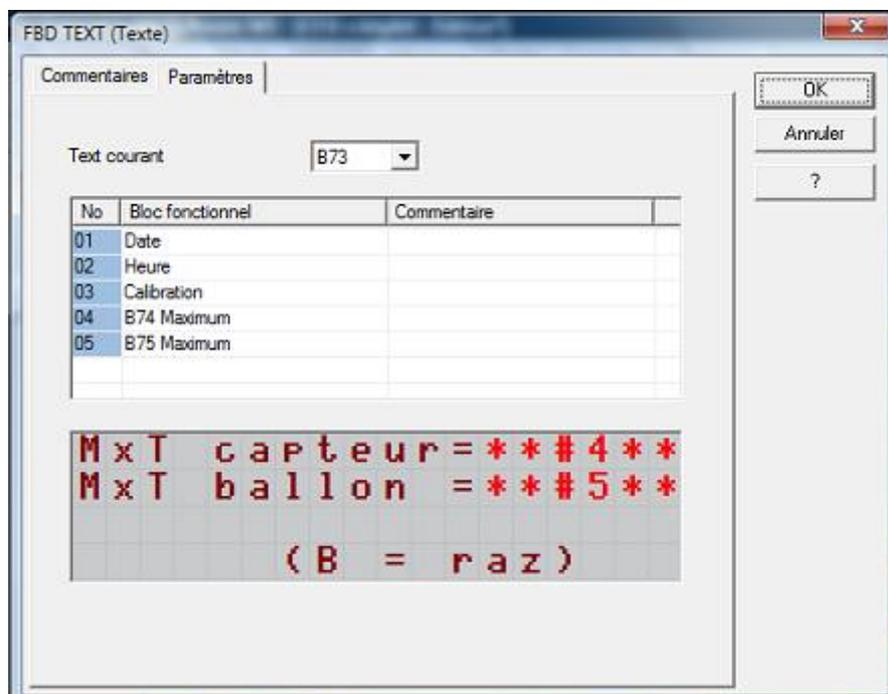
Sauvegarde sur coupure secteur
 Modification autorisée

Comme il s'agit d'un programme que vous pouvez prendre tel quel puis le modifier, voici le paramétrage des 4 blocs TEXT.

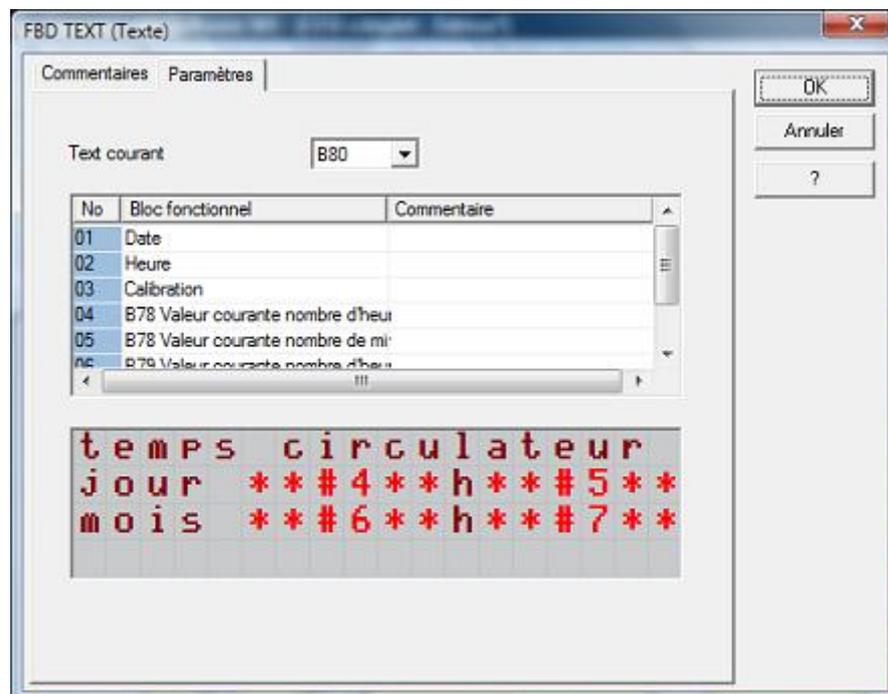
Le bloc TEXT de l'état initial, donc qui s'affiche par défaut : celui des températures (rappel : quand les données sont rouges, elles ne sont pas modifiables) :



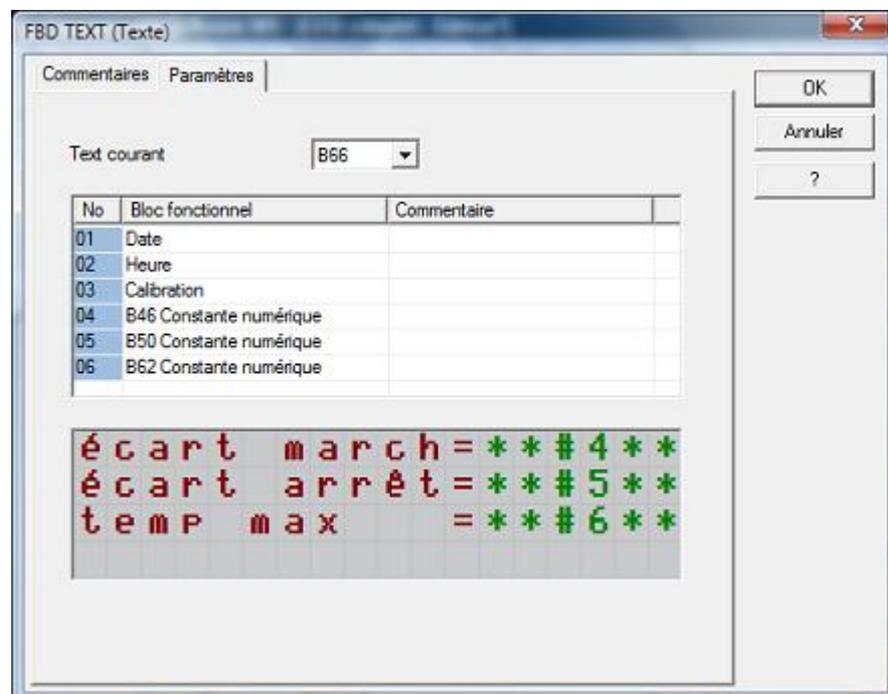
Le bloc TEXT affichant les températures maximales :



Le bloc TEXT affichant les cumuls de temps de fonctionnement :



Et enfin le bloc TEXT affichant les paramètres et permettant leur modification (rappel : quand les données sont vertes, elles sont modifiables dans l'écran) :



A noter : ce programme occupe environ :

- 20 % de la capacité mémoire d'un Millenium de type CD12 (8 entrées/4 sorties) ou CD20 (12 entrées/8 sorties)
- 10 % de la capacité mémoire d'un Millenium de type XD26 (16 entrées/10 sorties, extensible)