15/99

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

ECOLE NATIONALE POLYTECHNIQUE

D. E. R. Génie Électrique et Informatique
Département d'Électronique

Projet de Fin d'Études

Présenté et soutenu en vue de l'obtention de l'ingéniorat d'état en électronique

THEME

Simulation et contrôle d'une station de pompage photovoltaïque avec Visual Basic

Réalisé par :

BOUHEBBAL OUALID.

Proposé par :

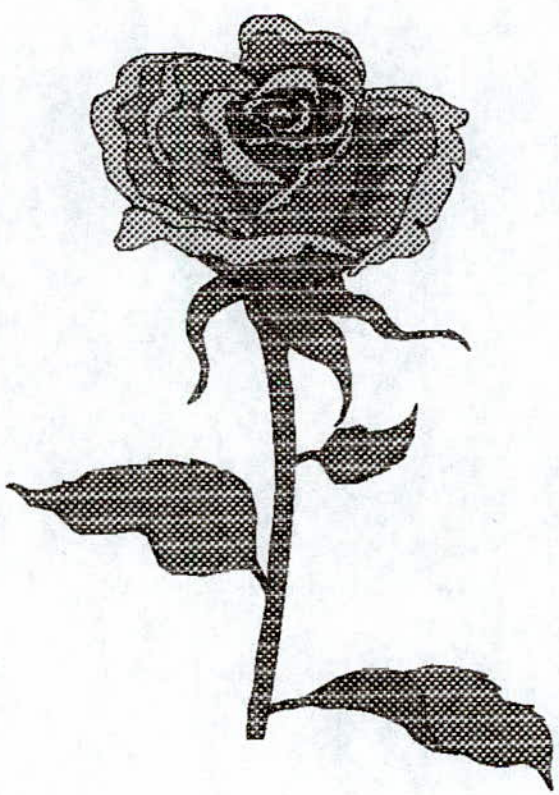
M. **HADDADI M.**

Session
Septembre 1999

Je dédie ce modeste travail à l'âme qui nous a quittés à jamais...

À la fleur qui ne fane à plus jamais...

et qui restera pour toujours dans notre cœur.



β. ω.

ملخص

هذا المشروع يتمثل في برنامج لمحاكاة تشغيل محطة ضخ بالطاقة الشمسية. و يهدف البرنامج إلى مراقبة محطات موجودة فعلا أو تحديد كمي قصد تصميم محطات جديدة مع مراعاة مختلف أنواع المحطات الممكنة.

وقد استُعمِل لإنجازه لغة Visual Basic مما يضفي عليه واجهة حوار مكنف مع المستعمِل تمكنه من إضافة أو إلغاء المركبات التي يريد بكل سهولة.

الكلمات المفتاحية: محاكاة، نموذج، مضخة، مموّج، مدخّرة، لوحة شمسية، تحديدكمي، برنامج.

Résumé

Ce projet consiste en la réalisation d'un programme de simulation de station de pompage photovoltaïque en vue de leurs surveillance et dimensionnement, ceci quelque soit le type de station choisi par l'utilisateur.

Ce programme a été écrit avec Visual Basic et présente une interface interactive très agréable permettant à l'utilisateur le choix libre des éléments de la station.

Mots clés : Simulation, modèle, pompe, onduleur, batterie, panneau solaire, dimensionner, programme, Visual Basic, Windows 32 .

Abstract

In this work, we simulate photovoltaic pumping stations. We use Visual Basic which gives an interactive interface with the user who can add or éliminate any part of the station.

The object is not only simulation, but also checking or dimensionning the station as well.

Key words : Simulation, model, pump, DC-AC converter, batteries, solar pannel, dimensionning, program, Visual Basic, Windows 32 .

Sommaire

INTRODUCTION

Dans le monde entier, tous les experts reconnus recommandent le passage à un approvisionnement énergétique basé de plus en plus sur le rayonnement solaire. Dans les conditions actuelles du marché, l'utilisation du solaire thermique, notamment dans les nouvelles constructions, permet la substitution d'une part considérable d'énergie fossile, alors que la contribution d'électricité solaire reste extrêmement modeste, pour des raisons de coûts.

Malgré les progrès, nos rapports actuels avec l'énergie, et la tendance qui en découle, restent cependant incompatibles avec les exigences d'un développement durable.

Aujourd'hui, l'économie, dépendant mondialement à 90 % de l'énergie fossile, produit le double de CO_2 de ce qui serait tolérable, si la charge de l'écosystème global ne devait pas être dépassée à long terme. Le professeur A. Zuberbühler, Président de la commission suisse de l'énergie avait dit à ce propos "*Nous pouvons entreprendre les changements indispensables, si nous nous décidons à nous tourner vers les sources de prospérité renouvelables : l'énergie solaire et l'intelligence humaine*".

Effectivement, un gros effort de recherche est encore nécessaire aussi bien pour la technique des centrales thermiques solaires que pour la technique photovoltaïque afin de rendre attractif ce type d'énergie.

Grâce à sa structure modulaire, la technique photovoltaïque est surtout adaptée à une utilisation décentralisée nécessitant une puissance réduite. Parmi les avantages de cette technique, il faut signaler sa bonne intégration dans les structures des bâtiments qui évite de dénaturer visuellement l'environnement. De plus les collecteurs sans installation de focalisation transforment non seulement le rayonnement direct, mais également le rayonnement diffus ; enfin les installations sont robustes, laissant prévoir une longue durée de vie.

Les inconvénients de cette technique sont le coût élevé des cellules et leur faible rendement. Dans les installations couplées à un réseau, le courant continu doit être transformé en alternatif par un convertisseur ; les installations autonomes nécessitent un accumulateur pouvant couvrir les besoins d'une semaine environ.

Le prix de revient du courant produit par une installation photovoltaïque est environ dix fois supérieur au prix actuel de l'électricité. Dans la technique prédominante et déjà bien

avancée du silicium mono- ou polycristallin, aucune percée qui permettrait une baisse massive des coûts n'est en vue pour le moment. L'augmentation de la production au cours de ces dernières années n'a pas non plus entraîné une baisse de prix radicale. Des progrès sont possibles dans d'autres technologies cellulaires, mais la recherche en est encore à un stade précoce. Les limites physiques des convertisseurs photovoltaïques ne sont pas encore atteintes.

En ce qui nous concerne, et parallèlement à ces considérations "écologiques", il faudra aussi tenir compte que l'approvisionnement en électricité dans les régions rurales isolées est un problème d'actualité. L'extension du réseau pour des demandes relativement faibles et isolées n'est pas rentable pour SONELGAZ. Bien que la solution des groupes électrogènes (Diesel) présente beaucoup d'inconvénients (peu fiables, peu autonomes, coûts cachés pour le combustible, les réparations et l'entretien), elle a souvent été choisie pour son coût d'investissement modéré. En effet, le coût initial élevé d'un générateur photovoltaïque est l'obstacle majeur à son expansion sur ce type de marché, en particulier dans les pays en voie de développement où les taux d'intérêt sont souvent très élevés.

Les dizaines de milliers d'unités photovoltaïques autonomes (au silicium monocristallin, polycristallin ou même amorphe) installées de par le monde ont pourtant démontré leur compétitivité en ce qui concerne de multiples applications de petite et moyenne puissance (inférieur à 10 kW).

De nombreuses organisations internationales d'aide aux pays en voie de développement ont choisi la technologie photovoltaïque comme outil de développement social et économique pour fournir des services de base à la population, tels que :

- le pompage de l'eau pour la consommation du village ou pour l'irrigation,
- la réfrigération pour la production de glace et la conservation de vaccins, sang, produits agricoles,...,
- l'éclairage (lampe portative, éclairage public, électrification villageoise,...)

C'est alors dans cet esprit que nous avons initié ce travail qui consiste à élaborer un programme informatique d'aide à la conception et au suivi d'un système de pompage photovoltaïque.

Ce programme se veut un outil qui permettra à un utilisateur non-spécialiste d'avoir toutes les informations (coût équipements y compris) concernant l'installation de pompage

qu'il désire mettre en œuvre. Nous avons aussi prévu la possibilité d'une surveillance de l'installation réalisée en adjoignant au micro-ordinateur une carte d'acquisition adéquate.

Ce programme a été écrit en Visual Basic afin d'offrir une interface agréable à l'utilisateur.

Nous allons donc exposer ce travail selon les points suivants :

- Description d'un système de pompage photovoltaïque
- Modélisation de ses éléments
- Dimensionnement des éléments utilisés et contrôle (surveillance)
- Élaboration du programme de gestion
- Conclusion

CHAPITRE I

DESCRIPTION

I. Description d'un système de pompage photovoltaïque

Dans un système de pompage quelconque, le but à atteindre est de soulever un liquide à une hauteur donnée et avec un débit déterminé.

Pour ceci, il faut avoir une pompe convenable.

L'animation de la pompe nécessite un moteur qui peut être :

- Soit thermique (à explosion ou diesel),

Soit électrique (dans ce cas il est soit à courant continu ou alternatif).

Le moteur est lié mécaniquement à la pompe :

- Soit directement avec un même arbre,
- Soit à travers un système de transmission de mouvement (poulies, engrenages...).

Le pompage du liquide augmente son énergie potentielle (qui se transforme en mouvement à l'utilisation) et selon le principe de conservation de l'énergie qui doit satisfaire les critères suivants :

- Alimentation continue sans arrêt durant toute ou une partie de la journée,
- Puissance assez grande pour couvrir la puissance nécessaire au soulèvement du liquide, plus des pertes émanantes dans différentes liaisons entre les composantes du système,
- Un prix de revient pour le m^3 minimum,
- Adaptation à la conversion en mouvement (on exclut certaines formes d'énergie qui ne peuvent être accommodés au système).

Dans un environnement rural ou montagneux par exemple, l'énergie solaire satisfait à ces critères avantageusement par rapport aux autres formes d'énergie (réduction des frais de transport et d'alimentation).

Dans ce qui suit, nous allons passer en revue les différentes composantes d'une station de pompage pour en sortir sur une classification méthodique des différentes configurations.

I.1. Le panneau

Un panneau solaire est un assemblage d'unités élémentaires appelées cellules. Une cellule photovoltaïque est une jonction PN qui convertit l'énergie lumineuse en électricité.

Physiquement, les panneaux peuvent être fixes ou orientables ; et dans ce cas, il y a plusieurs possibilités :

- Orientation manuelle : elle s'effectue périodiquement (par exemple durant les tournées de vérification hebdomadaire, mensuelle ou saisonnière) et elle ne peut donner des résultats optimaux, d'une part à cause de la nécessité d'une main d'œuvre expérimentée d'autre part à cause de la complexité mécanique des supports.
- Orientation avec un système d'horlogerie qui peut être alimenté directement par le panneau et son système de stockage via un moteur électrique.

Ce système est coûteux à cause de sa mécanique et de sa consommation d'énergie durant la nuit sans rendement (pour ramener le système à se diriger vers l'Est, le matin).

- Orientation par un moteur guidé par un système intelligent (un microprocesseur ou un microcontrôleur) : cette solution relativement récente semble être plus économique que les précédentes malgré les coûts initiaux du système qui tendent à baisser sans cesse avec la disponibilité du matériel et du logiciel nécessaire.

Ce type de guidage fournit l'optimum en ce qui concerne les performances des panneaux solaires et, en plus, nécessite un minimum d'intervention de l'opérateur humain.

Les cellules sont réunies en modules ; qui à leur tour sont réunis en panneaux ; et finalement, les panneaux sont des unités physiquement indépendantes qu'on peut assembler en champs (comme indiqué à la figure I-1).

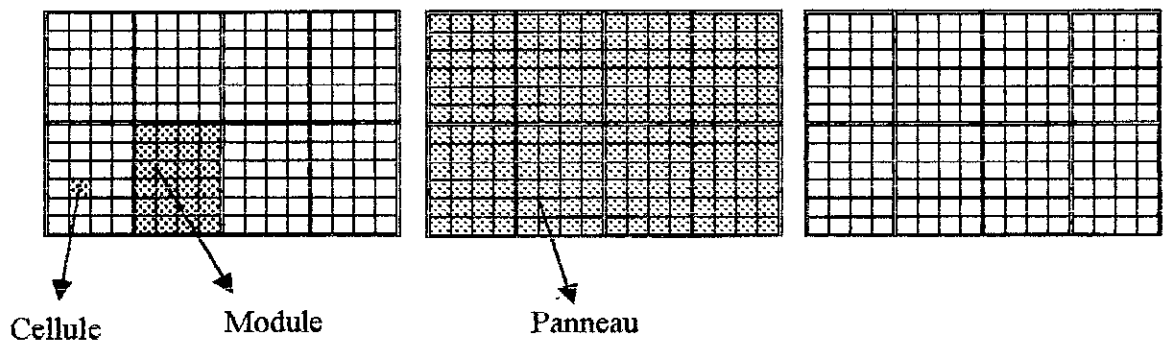


Figure I-1 : Structure d'un champ photovoltaïque.

A la sortie, on peut varier à volonté les caractéristiques du générateur (tension et courant) par la mise en série et en parallèle d'un nombre convenable de panneaux. Ceci ne doit pas en principe poser de problèmes si l'irradiation du panneau reste constante ; mais quand on fait le calcul pour un certain éclairement et qu'on détermine le nombre de panneaux nécessaires (en série et en parallèle), le problème est que l'ensoleillement varie durant la journée et ainsi ces caractéristiques varient et il serait difficile, avec une charge fixe, d'avoir toujours le même rendement.

On fait, alors, recours à des systèmes qu'on appelle systèmes d'adaptation d'impédance (par analogie avec le processus d'adaptation d'impédance dans les circuits linéaires).

La méthode la plus rentable est que ces circuits exécutent en même temps une tâche différente, le plus souvent une conversion DC-DC ou DC-AC.

Dans notre projet, nous avons besoin des deux types de convertisseurs : on utilisera le premier pour la charge de la batterie et le second pour l'alimentation du moteur au cas où il s'agirait d'un moteur à courant alternatif.

Dorénavant, on considère la batterie avec son système d'adaptation (y compris le convertisseur DC-DC) comme un ensemble unique.

I.2. Description du système du stockage (les batteries)

La nécessité d'un système pour le stockage de l'énergie s'impose du moment où la source d'énergie qui est le soleil n'est pas disponible à tout moment, tandis que le besoin de cette énergie peut se sentir pendant que le soleil est absent (la nuit). Celle-ci n'est pas la seule raison, car en plus, on peut demander une puissance de pointe en des moments où l'ensoleillement est faible. citons deux exemples :

Dans le cas de l'irrigation dans les régions sèches (désert et hauts plateaux), il est demandé de fournir toute la puissance, le soir (l'irrigation du soir garantie la conservation de l'humidité dans le sol pour une longue durée). Il faut, alors, stocker de l'énergie durant toute la journée pour la restituer au soir.

Dans les systèmes qui demandent une autonomie (exemple : dans les unités de distribution ou de production industrielle) surtout à la survenue de temps couvert (mauvais temps).

Il existe plusieurs façons de stocker l'énergie, sous forme de chaleur, de pression ou sous forme chimique. Le cas qui nous est le plus favorable est le dernier quand il s'agit de batteries d'accumulateurs car on la restitue sous forme électrique.

I.2.1. Fonctionnement de batteries

Il existe deux types principaux de batteries ; des batteries au plomb et celles au nickel-cadmium.

Les batteries au plomb sont constituées de plaquettes de deux types différents de matériau : du plomb au zinc et de l'oxyde de plomb. Les plaquettes de types différents sont disposées alternativement et séparées par des feuilles en matière isolante et résistante aux acides (leur rôle est d'empêcher les courts-circuits) puis elles sont montées dans des récipients, généralement étanches, remplis d'acide sulfurique dilué.

A la charge, l'acide réagit avec l'oxyde de plomb (en réalité, l'acide joue le rôle de catalyseur et c'est l'eau qui réagit effectivement avec l'oxyde de plomb), alors, il se forme de l'hydrogène qui est absorbé par les plaquettes poreuses d'oxyde de plomb. Cet hydrogène constitue la véritable réserve d'énergie.

Pendant la décharge, une réaction inverse s'effectue entre l'hydrogène et l'oxyde de plomb en restituant l'énergie sous forme de courant électrique.

La première charge de la batterie se fait dans l'usine car elle nécessite une concentration de l'acide très précise et une graduation de courant particulière. Son rôle est de préparer les matériaux pour les prochaines charges et décharges.

I.2.2 Caractéristiques principales des batteries

Il existe deux types de batteries qui sont les plus répandues dans le monde : Les batteries au plomb (Pb) et celles au nickel–cadmium (Ni–Cd) ; ces dernières possèdent de meilleures caractéristiques (voir tableau 1) mais elles sont plus chères.

	Pb	Ni–Cd
Robustesse	Normale	Très bonne
V_{nom} (volts)	1,7-2,45 change avec la charge	1,25 constante avec la charge
$R_{interne}$	Très faible	Plus grande
η (%)	70-80	60-70
Durée de vie (ans)	7-10	8-10
Nombre de cycles pour 80% de charge	1000-1500	1500-2000
Surcharges et décharges profondes	Mal acceptées	Acceptées (5-20%)
Auto décharge	Faible (3-5% par mois)	Faible
Charge	Lente (à faible régime)	Régimes élevés
Effet de la température	Mauvaises performances aux basses températures	Fonctionnement normal sur une large plage de température
Entretien	Nécessaire (régulier)	Peu (néant si étanche)
Risque de projection	Projection d'H_2SO_4	Projection d'une solution basique dangereuse
Coût	Faible	Elevé

La caractéristique la plus importante d'une batterie est sa capacité. Elle s'exprime en Ampères-heures (A-h) et représente la durée nécessaire (en heures) pour vider une batterie si l'on en tire un courant de 1 Ampère (et si on tire x Ampères, on aura une durée de $\frac{C}{x}$ heures).

I.3. La pompe

Dans le domaine de pompage de l'eau, les pompes les plus fréquemment utilisées sont les pompes centrifuges à cause de caractéristiques mieux adaptées. Pour cette raison nous y consacrons l'essentiel de ce chapitre.

Dans certains cas particuliers, on peut être amené à utiliser des pompes élévatrices d'un autre type (vis d'Archimède, pompe à piston... etc.).

I.3.1. Constitution d'une pompe centrifuge

Un distributeur : sorte de tubulure profilée qui, comme son nom l'indique, sert à conduire l'eau avec une vitesse et une direction convenables dans l'axe de la pompe ou «ouïe » ou «œuillard » de la roue.

Il est généralement constitué par un cône convergent qui permet de réaliser une meilleure disposition des filets liquides en améliorant le parallélisme et l'égalité de vitesse. Il est précédé à l'avant par la canalisation d'aspiration.

Une roue (ou turbine ou rotor ou, encore, mobile) : constituée par un moyeu porté par un arbre et muni d'aubes tournant à l'intérieur de deux coquilles formant le corps de la pompe.

Les aubes peuvent être fixées sur un ou deux côtés à des disques ; on distingue ainsi les rotors ouverts, semi-ouverts ou fermés (figure I-2)

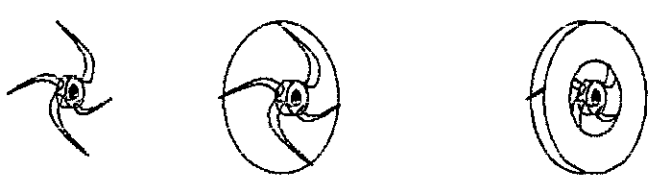


Figure I-2 : Rotors de pompe

Un diffuseur : qui peut être :

- Lisse,
- A ailettes,
- En limaçon.

Un cône divergent : il se trouve à la fin du diffuseur, il sert à ralentir la vitesse du liquide de sorte que toute l'énergie qu'il acquière devient une énergie potentielle de pression.

I.3.2. Fonctionnement

Le liquide entre par le centre de la pompe à travers le convergent sous forme de filets de courant poussé par la pression atmosphérique externe. A l'atteinte de l'hélice, il subit une poussée vers les extrémités causée par la forme de celle-ci pendant sa rotation (il ne s'agit donc pas proprement de force centrifuge comme le nom l'indique). La vitesse et la pression du liquide à la sortie de l'hélice dépendent du rayon de celle-ci, de sa vitesse de rotation et de sa forme.

A la sortie le liquide est recueilli par le divergent qui le dirige vers le canal de refoulement sous forme, toujours, de filets.

Pour un fonctionnement normal de la pompe, le liquide pompé doit parvenir d'un endroit non clos car, en effet, c'est la pression de l'air qui l'amène jusqu'à l'hélice. Ceci explique le fait que le pompage soit impossible si la hauteur de la pompe est supérieure à 10m au-dessus de la surface libre du liquide. Une disposition idéale est d'immerger la pompe complètement dans le liquide.

I.4. Les différents types de stations

I.4.1. Station très simple : (figure I-3)

Figure I-3 : Pompage au fil du soleil

I.4.2. Station simple avec conversion DC-AC : (figure 1-4)

Figure I-4 : Station simple avec conversion DC-AC.

I.4.3. Station simple avec stockage : (figure I-5)

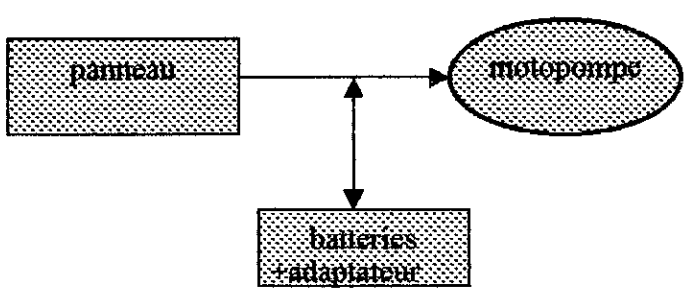


figure I-5 : Pompage au fil du soleil avec stockage

I.4.4. Station avec stockage et conversion DC-AC :

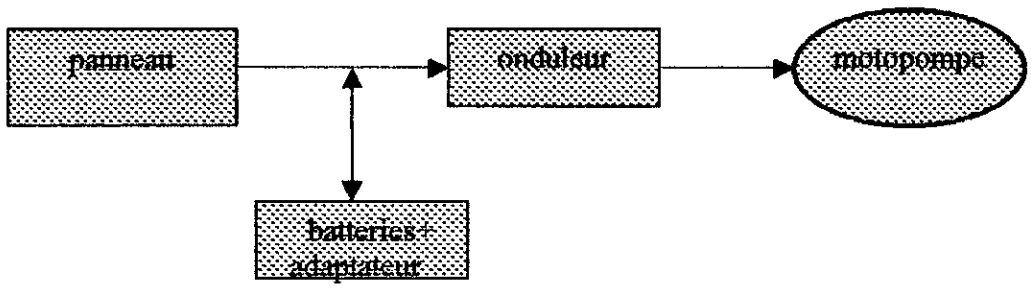


Figure I-6 : Station de pompage avec stockage et conditionnement du courant.

I.4.5. Station complète :

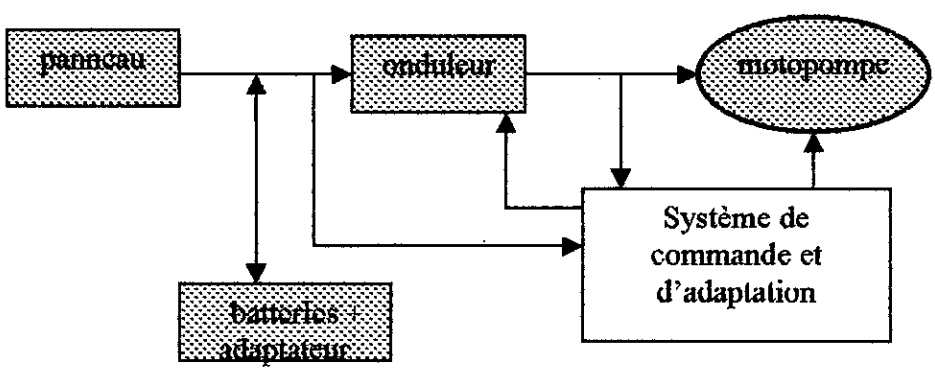


Figure I-7 : Station de pompage avec stockage et conditionnement du courant.

CHAPITRE II

MODELISATION

II. Modélisation

À l'heure actuelle, c'est dans les endroits isolés, loin d'un réseau électrique et où les besoins en énergie sont relativement faibles (généralement moins de 10 kW_p) que l'énergie photovoltaïque est la plus concurrentielle. Dans ce type de site, l'énergie solaire est captée par les modules qui serviront dans certains cas à recharger des accumulateurs. L'énergie électrique ainsi emmagasinée permet alors de répondre à la demande d'électricité. Le pompage de l'eau est également une application intéressante, moins répandue cependant. Cette application a l'avantage de permettre éventuellement le stockage de l'énergie solaire sous la forme d'une simple réserve d'eau accumulée pendant les heures d'ensoleillement, plutôt que par des accumulateurs électriques. Dans les sites isolés hors-réseau, l'utilisation du PV est très concurrentielle par rapport à l'extension du réseau, aux batteries non rechargeables, et aux génératrices à effet Peltier, à moteur diesel ou à essence. Aux Etats-Unis, selon les estimations de l' Utility Photovoltaic Group (UPVG), il en coûte de 12 000 à 50 000 $ du km pour étendre un réseau. En Algérie, on pense que le photovoltaïque devient concurrentiel dès que le réseau est éloigné de plus de 60 km. Le photovoltaïque est donc une excellente option dans le cas des demandes faibles et dans les régions éloignées du réseau des services publics. Par comparaison avec les génératrices diesel et les batteries non rechargeables, le grand avantage du PV est la réduction des coûts d'exploitation, de maintenance et de remplacement, ce qui se traduit souvent par une baisse des coûts du cycle de vie des systèmes PV.

C'est dans ce contexte que nous avons élaboré notre programme qui servira à dimensionner les éléments d'une station de pompage autonome, avec ou sans stockage électrique.

Nous allons donc dans ce chapitre commencer par décrire les modèles utilisés pour les éléments de ces stations. La description du programme informatique s'effectuera dans le chapitre suivant.

Un des principaux avantages du programme est de permettre une espèce d'analyse de sensibilité pour déterminer les paramètres essentiels à la viabilité financière d'un projet. L'analyse de sensibilité permet de faire varier les valeurs entrées par l'utilisateur manuellement dans les fenêtres appropriés, Analyse des coûts et Sommaire financier. Dans la plupart des cas, les paramètres financiers, comme "Coût évité - Energie", "Coût évité - Puissance", "Inflation" et "Indexation - Coût de l'énergie", peuvent avoir une grande influence sur la rentabilité globale d'une centrale donnée.

En plus de ces paramètres, l'utilisateur désirant effectuer une analyse de sensibilité devrait prendre d'abord en considération les paramètres suivants :

Inclinaison et orientation du capteur solaire lorsque la feuille Ressource solaire est utilisée ;

Rendement nominal du champ photovoltaïque et rendement du conditionneur d'énergie ;

Coûts des modules PV et des onduleurs ; et

Coûts imprévus.

Nous passons en revue les différents modèles pour chaque élément de la station de pompage :

II.1. Le panneau

Le panneau solaire étant la source d'énergie de notre station, il faut lui attribuer une grande importance dans la simulation et le contrôle.

La grandeur fondamentale à laquelle on s'intéresse est la puissance p =I.V fournie par le panneau. Elle doit être maximale.

De toutes façons, la puissance fournie est fonction de la tension et du courant débité qui se déterminent par le point de fonctionnement.

Ce point de fonctionnement est le point de rencontre de la caractéristique I = f (V) du panneau avec celle de la charge.

Notre première occupation sera de déterminer la caractéristique I(V) qui dépend à la fois du panneau et aussi de son environnement.

La caractéristique de la charge peut être fixe (cas de la charge résistive) ou bien variable selon les conditions initiales (si la charge contient des capacités ou des inductances...etc.).

II.2. Détermination des caractéristiques du panneau :

Vu que l'énergie reçue par le panneau dépend de l'éclairement " E " qui varie dans le temps, en général, de manière aléatoire (les variations des conditions climatiques) ; toute la caractéristique I (V) change avec l'éclairement.

D'un autre côté, une cellule solaire étant un dispositif à semi-conducteurs, sa caractéristique est très sensible aux variations de la température de sa jonction T_j qui est fonction, à la fois, de la température ambiante T_a et de l'éclairement E qui contribue à son échauffement.

Une relation empirique est donnée par :

$$T_j = T_a + 30\,E \quad \text{.....(II-1)}$$

Où :

T_j : température de la jonction en ° C.

T_a : température de la jonction en ° C.

E : éclairement en kW/m².

Plusieurs modèles ont été proposés pour la simulation de la caractéristique du panneau. Nous n'en citerons que deux :

II.2.1. Modèle de Singer :

Ce modèle de la caractéristique I = f (V) pour une température et un éclairement donnés, tient compte uniquement de trois caractéristiques qui peuvent être mesurées dans des conditions typiques de température ambiante et d'éclairement. ce sont :

La tension de circuit ouvert : V_{co}.

Le courant de court-circuit : I_{cc}.

La puissance maximale : P_m.

La généralisation étant possible ensuite grâce aux lois suivantes :

La tension de circuit ouvert : V_{co} varie selon la loi :

$$V_{co} = V_{coo}\,(1 - \gamma\,\Delta T)\,\mathrm{Ln}\,(e + \beta\,\Delta E) \quad \text{.....(II-2)}$$

Où :

V_{coo} : La valeur de V_{co} à la température de référence,

γ : Coefficient de température de la tension de circuit ouvert,

ΔT : Différence entre la température actuelle et celle de référence,

e : Base du logarithme népérien e ≈ 2,718281828459,

β : Coefficient d'influence de l'éclairement,

ΔE : Différence entre l'éclairement actuel et celui de référence.

Le courant de court circuit : I_{cc} varie selon la loi :

$$I_{cc} = I_{cco} (1 + \alpha \Delta T) E \quad \text{(II-3)}$$

Où :

I_{cco} : La valeur de I_{cc} à la température de référence,

α : Coefficient de température de la tension de circuit ouvert,

ΔT : Différence entre la température actuelle et celle de référence,

E : L'éclairement actuel

La puissance maximale : P_m varie, approximativement, selon la loi :

$$P_m = P_{mo} . \frac{V_{oc} . I_{cc}}{V_{oco} . I_{cco}} \quad \text{(II-4)}$$

Le modèle se réfère au schéma suivant :

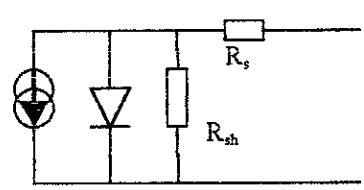


Figure II-1 : Schéma du modèle d'un panneau solaire.

R_s : résistance série.

R_{sh} : résistance de shunt.

En tenant compte que R_{sh} est très grande en général, on la néglige (pour se rappeler, elle est en parallèle), on a :

$$I - I_l - I_o \left[exp\left(\frac{V + R_s I}{A k T / e} \right) - 1 \right] \quad \text{(II-5)}$$

qu'on peut écrire, compte tenu que $exp\left(\frac{V + R_s I}{A k T / e} \right) >> 1$ ..... *(II-6)*

Il vient :

$I = I_l - I_o \exp \lambda(V + R_s I)$ ..... (II-7)

On a donc 4 inconnues I_l, I_o, λ et R_s.

Par contre, on a seulement 3 données V_{oc}, I_{cc} et P_m (ce qui donne 3 équations)

→ Le problème ne peut se résoudre qu'en éliminant une inconnue.

Et pour ce faire, il suffit de constater que pour un type de semi-conducteur donné avec une densité de dopage déterminée, on admet que le courant inverse de la jonction I_o est proportionnel à I_l et $I_o/I_l \approx 10^9$ pour le Si.

Cherchons alors les autres inconnues :

Pour I_l :

à $V = 0$, $I=I_{cc} \Rightarrow I_{cc} = I_l - I_o \exp \lambda(R_s I_{cc}). \approx I_l$ ..... (II-8)

$\Rightarrow$ $\boxed{I_l = I_{cc}}$ ..... (II-9)

$\Rightarrow$ $I = I_{cc} - I_o \exp \lambda(V - R_s I)$ ..... (II-10)

Pour λ :

à $I = 0$, $V = V_{oc} = (1/\lambda)\ Ln(I_{cc}/I_o)$ ..... (II-11)

$\Rightarrow \lambda = \frac{1}{V_{oc}} Ln \frac{I_{cc}}{I_o} \approx \frac{20,7}{V_{oc}}$. ..... (II-12)

On obtient, alors une représentation du type $V=f^{-1}(I)$ t.q. :

$$V = V_{oc}\left[1 + \frac{1}{20,7} \ln \frac{Icc - I}{Icc}\right] - Rs\, I \quad \text{..... (II-13)}$$

Pour R_s :

On cherche à identifier la puissance maximale obtenue par :

$$P_{max} - \max (V * I) - \max \left[V_{oc}\left(I + I \frac{1}{20,7} \ln \frac{Icc - I}{Icc}\right) - Rs\, I^2\right] \quad \text{..... (II-14)}$$

avec la valeur mesurée P_m.

On retient finalement la représentation donnée par l'équation n° II-14.

II.2.2. Modèle de Rauschenback :

Ce modèle nécessite la mesure de cinq paramètres :

Le courant de court-circuit : I_{cc},

La tension de circuit ouvert : V_{co},

Le courant à la puissance maximale : I_{mp},

La tension à la tension maximale : V_{mp},

La résistance série : R_s.

Dans ce cas, la caractéristique qu'on obtient pour des conditions typiques de température et d'éclairement est :

$$I = I_{cc}\left[1 - C_1\left(\exp\left(\frac{V}{C_2\ V_{co}}\right) - 1\right)\right] \quad \text{(II-15)}$$

où :

C_1 et C_2 sont des constantes données par :

$$\begin{cases} C_1 = \left(1 - \dfrac{I_{mp}}{I_{cc}}\right)\exp\dfrac{-V_{mp}}{C_2\ V_{co}} \\ C_2 = \dfrac{\dfrac{V_{mp}}{V_{co}} - 1}{\ln\left(1 - \dfrac{I_{mp}}{I_{cc}}\right)} \end{cases} \quad \text{(II-16)}$$

Pour une différence de température ΔT et d'éclairement ΔE, la caractéristique se translate selon la loi :

$$\begin{cases} \Delta I - \alpha\left(\dfrac{E}{E_{ref}}\right)\Delta T + \left(\dfrac{E}{E_{ref}} - 1\right)I_{cc} \\ \Delta V = -\beta\,\Delta T - R_s\,\Delta I \end{cases} \quad \text{(II-17)}$$

Remarques :

Ce dernier modèle fait intervenir un paramètre inaccessible à la mesure, à savoir R_s. Il faut donc utiliser le modèle de Singer ou une évaluation approximative de ce paramètre.

Les équations permettent de déduire la caractéristique correspondant à des conditions quelconques de température et d'éclairement à partir d'une caractéristique type. Il n'est alors pas obligatoire d'utiliser les équations pour déterminer cette caractéristique type mais n'importe quelle méthode nous permet de l'obtenir et c'est particulièrement intéressant si nous utilisons des caractéristiques expérimentales.

Les variations de T sont aléatoires mais celles de E peuvent être simulées.

Pour en conclure avec le panneau donnons la forme de sa caractéristique I – F (V) :

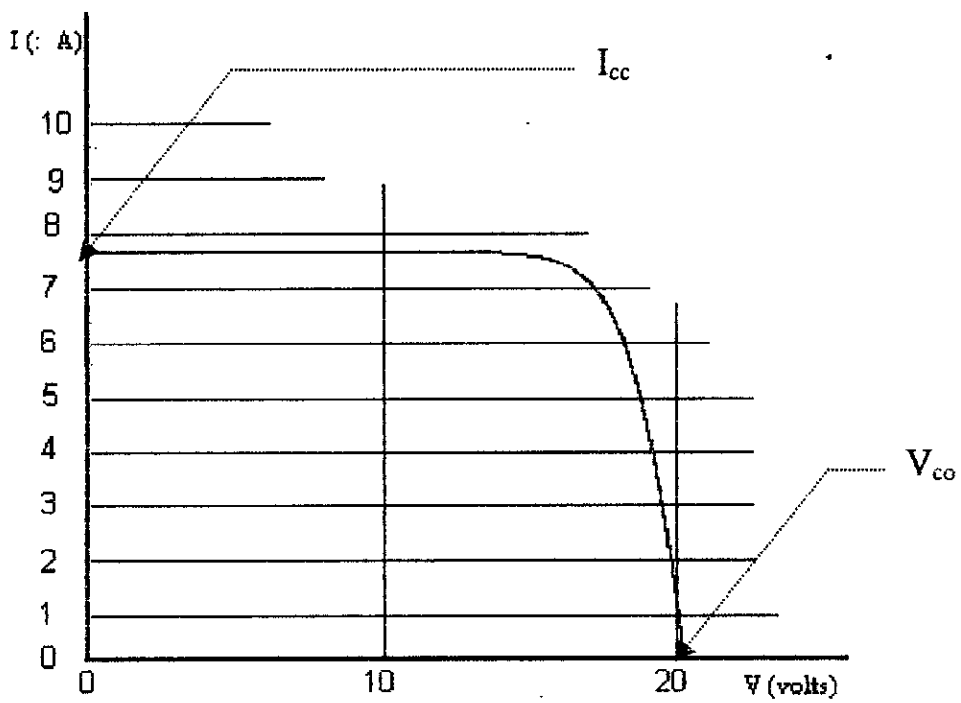


Figure II-2 : Caractéristique I = f (V) d'un panneau solaire.

II.3. L'éclairement

On peut distinguer deux types de l'éclairement qui agissent sur le panneau : l'éclairement direct et l'éclairement diffus.

II.3.1. L'éclairement direct

C'est le plus important en énergie par temps clair. Alors, et pour simplifier notre étude, nous nous contenterons de modéliser uniquement les beaux jours pendant lesquels le ciel n'est jamais couvert.

Il existe un angle d'ouverture défini par ses deux limites : matinale, crépusculaire ; qui dépendent du lieu (influence des montagnes, des immeubles... etc.) et de la saison.

Soit, alors, un panneau de surface S_0.

La puissance électrique totale reçue par ce panneau est :

$W_\perp = S_0 E$ ..... *(II-18)*

E(W/m²) étant l'éclairement global.

Si l'incidence de la lumière est oblique, la surface utile devient :

$S = S_0 \cos\alpha$ (α est l'angle d'incidence)..... *(II-19)*

D'où :

$E = S I = S_0 I \cos\alpha = S_0 I \vec{n} . \vec{i}$ ..... *(II-20)*

où :

$\vec{n}$: vecteur normal sur la surface du panneau et défini par les angles d'azimut et d'élévation (site) du panneau (φ et ψ)

$\vec{i}$: vecteur dirigé vers le soleil.

$\vec{n}$ est soit fixe soit variable selon que le panneau est fixe ou orientable, tandis que $\vec{i}$ est variable continuellement et doit être nul si le soleil n'éclaire plus le panneau (la nuit).

Le modèle suppose $\vec{n}$ donné (dans un repère lié au lieu géographique du panneau) et cherche à déterminer. Pour ceci nous devons effectuer une suite de changements de repère :

Le repère de base est un repère lié au centre de la terre et dirigé selon des directions fixes dans l'espace.

Le vecteur dirigé vers le soleil $\vec{i}$ est défini par :

$$[i_1]=\begin{bmatrix}\sin\theta\\ \cos\theta\\ 0\end{bmatrix}$$ où θ est proportionnel au n° du jour de l'année,. ..... *(II-21)*

Le 2ème repère équivaut à un repère dont l'axe Z_2 est incliné de 23°27' par rapport à Z_1, autour de X_1.

Z_2 est dirigé selon l'axe de rotation de la terre autour d'elle-même.

Le vecteur dirigé vers le soleil est, dans ce repère :

$[i_2] = [A_1].[i_1]$ où $[A_1]$ est la matrice de passage du 1er au 2ème repère.

$$[A_1]=\begin{bmatrix}1 & 0 & 0\\ 0 & \cos\beta & -\sin\beta\\ 0 & \sin\beta & \cos\beta\end{bmatrix}$$ où β = 32°27'. ..... *(II-22)*

Le 3ème repère a son axe X dans le méridien du lieu considéré.

Il est obtenu par une rotation du repère précédant autour de Z_2, d'un angle α proportionnel à l'instant considéré du jour.

Le vecteur dirigé vers le soleil est, dans ce repère :

$[i_3] = [A_2].[i_2] = [A_2].[A_1].[i_1]$ ..... *(II-23)*

où $[A_2]$ est la matrice de passage du 2ème au 3ème repère.

$$[A_2]=\begin{bmatrix}\cos\alpha & -\sin\alpha & 0\\ \sin\alpha & \cos\alpha & 0\\ 0 & 0 & 1\end{bmatrix}$$ où $\alpha=\left(\frac{heure}{24}+\frac{n^\circ\ du\ jour}{365}\right)*2\pi$ ..... *(II-24)*

Le 4ème repère a son axe Z_4 suivant la verticale du lieu considéré, X_4 est dirigé vers l'Est et Y_4 dirigé vers le sud.

Le vecteur dirigé vers le soleil est, dans ce repère :

$[i_4] =[i] = [A_3].[A_2].[A_1].[i_1]$ ..... *(II-25)*

où $[A_4]$ est la matrice de passage du 3ème au 4ème repère.

$$[A_3]=\begin{bmatrix}\sin\lambda & 0 & -\cos\lambda\\ 0 & 1 & 0\\ \cos\lambda & 0 & \sin\lambda\end{bmatrix}$$ où λ la latitude du lieu. ..... *(II-26)*

Il suffit de savoir $[i_1]$ pour déterminer $\vec{i}$; or $[i_1]=\begin{bmatrix}\cos\theta\\ \sin\theta\\ 0\end{bmatrix}$, θ donnée précédemment.

Conclusion :

L'éclairement direct est ainsi défini comme une fonction du temps et du lieu géographique qui sont connus (à tout moment).

II.4. Les batteries

Une batterie possède la caractéristique de charge représentée par la figure suivante

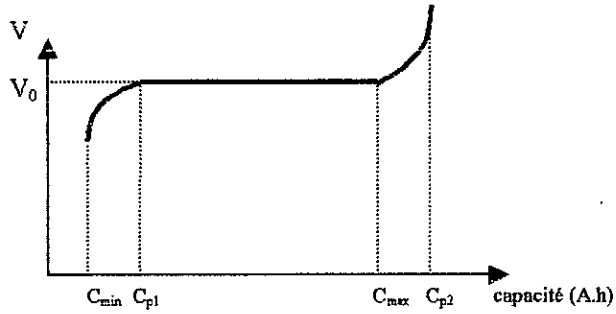


Figure II-2 : Courbe de charge d'une batterie d'accumulateurs.

Entre les valeurs C_{p1} et C_{p2} de la quantité d'électricité emmagasinée, la tension présente pratiquement un palier c. à. d. qu'elle reste constante.

Cette zone est la zone d'utilisation normale dans laquelle on peut charger ou décharger la batterie comme on veut (on s'efforcera donc à utiliser cette zone et à s'y maintenir).

Dans la zone entre C_{min} et C_{p1} quand on décharge beaucoup la batterie, la tension baisse au fur et à mesure ; ce qui est dû à la déshydrogénation partielle des plaquettes. Le processus (charge–décharge) reste pratiquement réversible dans cette zone mais si on décharge la batterie au deçà de (la charge minimale acceptable), alors, le processus devient irréversible (à cause de l'attaque des plaques par l'acide et leur destruction). Ainsi C_{min} est considérée comme étant une limite inférieure qu'il ne faut pas dépasser.

Dans la zone entre C_{min} et C_{p1}, on peut approximer la forme de la courbe à un polynôme de $3^{ème}$ degré ; ainsi :

$$V(C) = V_0 (C_{p1} - C)^3 \quad \text{.....(II-27)}$$

Dans la zone entre C_{p1} et C_{max} ; V augmente à mesure qu'on charge la batterie. Dans cette zone, on arrive à la saturation de la batterie (l'hydrogène n'arrive plus à s'adhérer aux plaques).

Si on continue à charger la batterie, on risque d'y avoir un dégagement rapide d'hydrogène avec un risque d'explosion ; c'est pourquoi on conseille de ne pas dépasser, en toutes circonstances 80% de la charge maximale.

Dans cette zone, on peut modéliser la courbe par l'équation :

$$V = V_0 \frac{C_{max} - C_{p1}}{C_{max} - C} \quad \text{(II-28)}$$

L'énergie emmagasinée étant $W = \int p\, dt$ *(II-29)*

$$W = \int V\, I\, dt \quad \text{(II-30)}$$

$$W = \int V\, (I\, dt) \quad \text{(II-31)}$$

$$W = \int V\, dC \quad \text{(II-32)}$$

L'énergie totale utile est : $W_T = \int_{C_{min}}^{C} V(C)\, dC$ *(II-33)*

Le passage de la charge de C_1 à C_2 augmente ou diminue alors l'énergie de :

$$\Delta W = \int_{C_1}^{C_2} V(C)\, dC \quad \text{(II-34)}$$

II.5. Les pompes

Dans ce qui suit, nous allons convenir sur les notations suivantes :

ϖ : poids volumique (N/m^3),

Q : débit (m^3/s),

r : rayon (m),

: vitesse angulaire de rotation de la roue (rad/s),

N : vitesse de la roue en (tours/mn),

$\vec{w}$: vitesse du liquide relativement à la roue,

$\vec{c}$: vitesse du liquide absolue, $\vec{c} = \vec{c}_u + \vec{c}_r + \vec{c}_z$,

$\vec{c}_m$: vitesse méridienne du liquide, $\vec{c}_m = \vec{c}_r + \vec{c}_z$,

$\vec{u}$: vitesse d'entraînement (vitesse de la roue).

On a :

$$\vec{c} = \vec{u} + \vec{w} \quad \text{(II-35)}$$

A l'entrée de la roue : $\vec{c}_1 = \vec{u}_1 + \vec{w}_1$, (II-36)

A la sortie de la roue : $\vec{c}_2 = \vec{u}_2 + \vec{w}_2$, (II-37)

L'application du théorème du moment cinétique conduit à :

Couple magnétique qu'il faut appliquer à l'arbre:

$$M = \frac{\varpi}{g} Q\left(C_{u_2} r_2 - C_{u_1} r_1\right) \quad \text{(II-38)}$$

Puissance fournie à l'arbre:

$$p = M\omega = \left(\frac{\varpi}{g} Qg\right) \frac{C_{U_2} u_2 - C_{U_1} u_1}{g} \quad \text{(II-39)}$$

où l'expression suivante est homogène à une hauteur :

$$H_{eff} = \frac{C_{U_2} u_2 - C_{U_1} u_1}{g} \quad \text{(II-40)}$$

Elle représente la hauteur à laquelle le liquide peut arriver avec le même débit Q si on lui fournit la puissance p.

Cette hauteur est toutefois idéale car elle ne tient pas compte des pertes de charge, et pour cela, on définit la hauteur nette $H_n = H_{eff} - \zeta_d - \zeta_r$, ... *(II-41)*

Où :

ζ_d = pertes de charge de diffusion par les aubages,

ζ_r = pertes de charge dues au frottement du liquide avec la roue.

Le rendement de la pompe : $\eta = \frac{H_n}{H_{eff}}$ ..... *(II-42)*

Le schéma suivant détermine la position des vitesses.

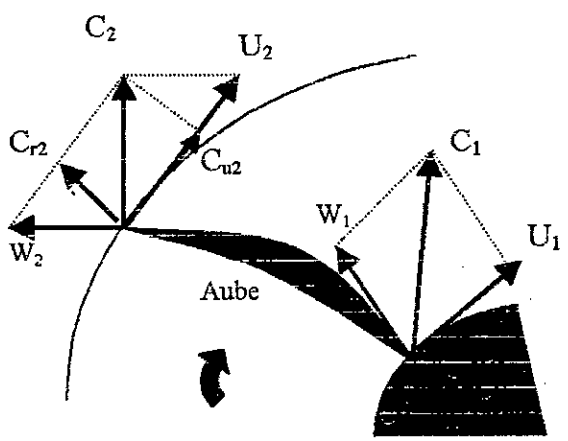


Figure II-3 : Schéma pour la disposition des vitesses dans l'hélice dune pompe.

On constate alors que pour un régime de fonctionnement déterminé par un débit Q et une vitesse de rotation N, les triangles de composition de vitesse à l'entrée et à la sortie sont parfaitement déterminés et de ce fait H_{eff} est bien déterminée.

Ce fait est une caractéristique qui différencie les pompes centrifuges des autres catégories.

En définitive, le fonctionnement d'une pompe centrifuge aux différents régimes possibles est caractérisé par une surface ayant pour équation :

$$F(H, Q, N) = 0 \quad \text{(II-43)}$$

A chaque point de cette surface correspond un point de fonctionnement de la pompe. RATEAU et BERGSON ont mis au point une représentation qui fixe l'une des trois variables H, Q ou N pour avoir une courbe bidimentionnelle des deux autres variables.
En général, on fixe N et on représente soit :

$H_{eff} = f(Q)$ (N est la vitesse de rotation de la roue),
$H_n = f(Q)$ à N constante,
$P = f(Q)$ à N constante,
$\eta = f(Q)$ à N constante,

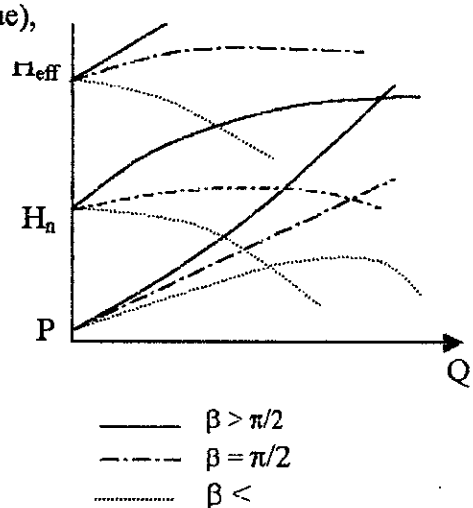


Figure II-5 : Caractéristiques d'une pompe.

On peut, de même, obtenir les représentations :

$Q = f(N)$ à H_{eff} constante,
$P = f(N)$ à H_{eff} constante,
$\eta = f(N)$ à H_{eff} constante,
$\eta = f(P)$ à H_{eff} constante,

La théorie des turbomachines nous donne alors :

$$H_n = N^2 + N Q \mu + k Q^2 \quad \text{(II-44)}$$

$$p = \frac{\varpi N Q_n}{g}\left(\mu_0 N - \frac{\lambda_0}{tg\ \beta} Q\right) \quad \text{(II-45)}$$

où μ_0 et λ_0 sont des fonctions de la dimension de la roue.

Remarque :

La hauteur manométrique effective à débit nul (H_{eff} à Q = 0) définit «la hauteur de barbotage ».

La courbe P = f (Q) ne passe pas par (0, 0) ; mais il y a une valeur initiale $p_0 = P|_{Q=0}$ due aux pertes considérables sous forme de chaleur qui provoque l'échauffement du liquide selon la loi indiquée par la figure suivante :

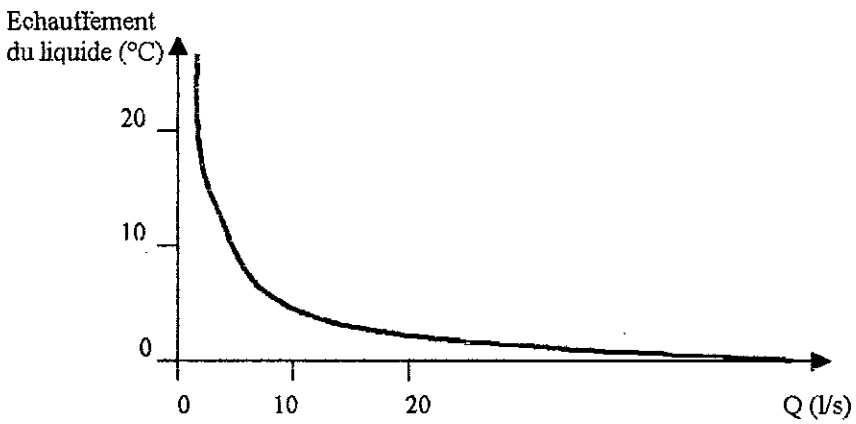


Figure II-6. Effet d'échauffement dans la pompe aux faibles débits.

II.5.1. Caractéristique du rendement :

On constata que le rendement varie très peu en fonction de la vitesse de rotation N.

Il reste pratiquement constant (si $N' = \frac{N}{2}$ alors $\eta' = \eta - 0.005$; $N' = \frac{N}{3}$ alors $\eta' = \eta - 0.01$)

$$\eta = \frac{\varpi Q}{g} H_n\left(\frac{1}{p}\right) \quad \text{.....(II-46)}$$

tel que $H_n = f(Q)$ et $P = f(Q)$.

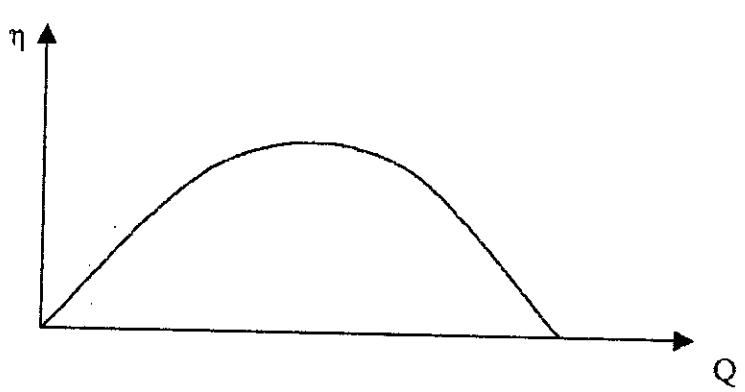


Figure II-7 : Caractéristique de rendement d'une pompe.

II.6. L'onduleur

Un onduleur est constitué principalement de paires de semi-conducteurs à commutation (généralement des MOS-FET de puissance) pour chaque fil de conduction (deux paires en monophasé et trois paires en triphasé).

Pour les très faibles puissances de la station (inférieurs à 1kW), on utilise le monophasé dont le schéma de principe est le suivant :

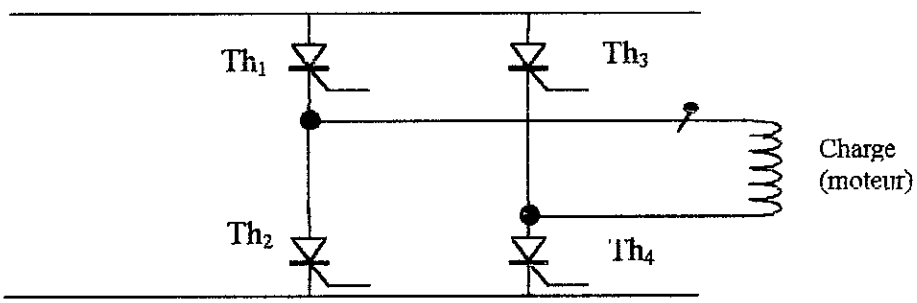


Figure II-8 : Onduleur monophasé (schéma de principe).

Le tableau de commutation est le suivant :

	Th_1	Th_2	Th_3	Th_4
Demi-période positive	Conducteur	Bloqué	Bloqué	Conducteur
Demi-période négative	Bloqué	Conducteur	Conducteur	Bloqué

Chaque thyristor conduit pendant une demi-période et s'éteint pendant l'autre.

Puisque la charge est inductive (moteur), on adjoint à chaque commutateur une diode de roue libre pour éviter les surtensions des commutations dues à la l'effet de loi de Lenz (figure II-9).

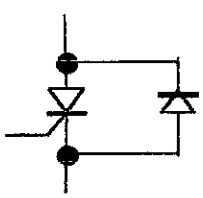


Figure II-9 : Thyristor avec diode de retour (ou de roue libre).

Dans le cas des plus grandes puissances (entre 1 kW et 10 kW), on utilise le triphasé (figure II-6)

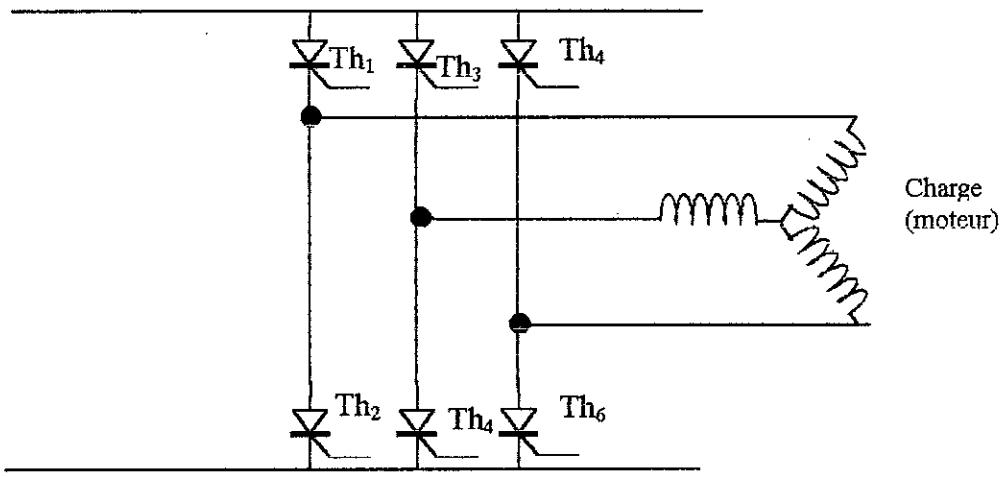


Figure II-9 : Onduleur monophasé (schéma de principe).

Le tableau des commutations est représente avec les courbes de phases :

	Th_1	Th_2	Th_3	Th_4	Th_5	Th_6
0 – T/6		×	×			×
T/6 – 2T/6		×	×		×	
2T/6 – 3T/6		×		×	×	
3T/6 – 4T/6	×			×	×	
4T/6 – 5T/6	×			×		×
5T/6 –T	×		×			×

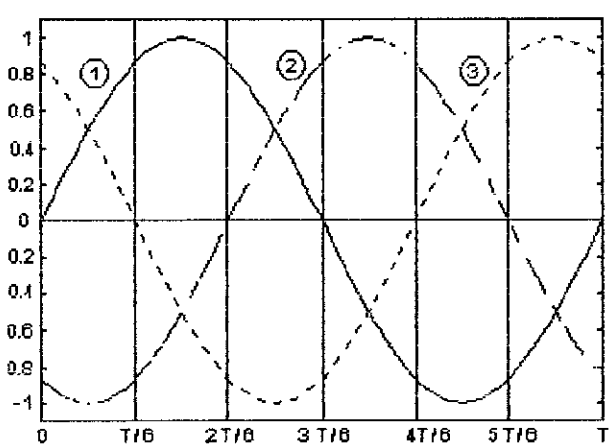


Figure II-10: Schéma des phases.

On constate toujours que chaque thyristor conduit pendant une demi-période ; mais ici, ils se déclenchent et s'éteignent indépendamment selon la séquence : 1-6-3-2-5-4.

Un des problèmes qui se posent dans la pratique est l'extinction (coupure) et le déclenchement des semi-conducteurs. Ainsi, on distingue trois types de commande :

- Conduction automatique, extinction forcée.
- Conduction forcée, extinction automatique.
- Conduction forcée, extinction forcée.

Alors, on adjoint aux schémas précédants des dispositifs adéquats pour l'extinction.

D'un autre coté, les moteurs électriques étant sensibles aux harmoniques, on adjoint aussi à chaque convertisseur un filtre pour les supprimer ; mais il y a une difficulté surtout dans notre cas où la fréquence de commande varie.

CHAPITRE III

DIMENSIONNEMENT CONTRÔLE

III. Dimensionnement et contrôle (surveillance)

III.1. But et principe du Dimensionnement

Avant d'entreprendre un projet pour une installation de pompage de ce type, il convient d'évaluer les besoins en matériel et d'estimer les coûts des différentes étapes de mise en œuvre. Dans ce but on fournit les besoins et les contraintes imposées par

III.2. Entrées

Les besoins sont exprimés en termes de hauteur manométrique totale (H_{mt}) qui est la somme de :

- La hauteur du réservoir par rapport à la terre H_1,
- La profondeur de la pompe H_2,
- Et les pertes de charges H_{pertes} avec $H_{pertes} = \alpha Q^2$.

Si le refoulement est sur un réservoir ouvert $\alpha \approx 0,1$ et sur un réservoir fermé $\alpha \approx 0,15$.

Donc

$$H_{mt} = H_1 + H_2 + H_{pertes} \quad \text{(III-1)}$$

Et en plus, il faut fournir le débit du liquide requis Q.

III.3. Calcul de la puissance hydraulique nécessaire

La puissance requise pour l'obtention du débit Q à la hauteur H_{mt} est :

$$P_{hyd} = \rho \cdot g \cdot Q \cdot H_{mt} \qquad [Q \text{ en } m^3/s] \quad \text{(III-2)}$$

Où :

ρ : masse volumique du liquide en [kg/ m^3]

$\rho_{eau} = 1000$ à 4°C.

III.4. calcul de la puissance nécessaire à la motopompe

la motopompe ayant un rendement η_{mp}, si elle reçoit P_{mp} elle fournit la puissance hydraulique P_{hyd} ce qui s'exprime ainsi :

$$P_{hyd} = \eta_{mp} \cdot P_{mp} \quad \text{(III-3)}$$

Soit : $P_{mp} = \frac{P_{hyd}}{\eta_{mp}} = \frac{\rho.g.}{\eta_{mp}} Q.H_{mt}$

Numériquement $P_{mp} = \frac{Q.H_{mt}}{0,367}$ .....(III-4) et(III-5)

Selon le type de la motopompe (à courant continu ou alternatif), on peut éventuellement faire intervenir le rendement de l'onduleur :

$$P'_{mp} = \frac{P_{mp}}{\eta_{conv}} = \frac{P_{hyd}}{\eta_{mp}.\eta_{conv}} \quad \text{.....(III-6)}$$

P'_{mp} (ou P_{mp} pour une motopompe à courant continu) est la puissance requise à l'ensemble module photovoltaïque + stockage éventuel.

Dans ce stade, le type de station détermine la démarche :

a) Pour une station sans stockage (au fil du soleil) :

Pour ce type de station, on considère que Q est la moyenne journalière requise et dans ce cas, il suffit de prendre pour la puissance photovoltaïque la moyenne journalière (puisque justement la puissance est proportionnelle au débit Q, sinon il fallait procéder autrement).

Alors, seul le rendement du module photovoltaïque η_{pv} intervient

$$P_{pvmoy} = Q_{moy} \frac{\rho.g.H_{mt}}{\eta_{pv}.\eta_{conv}.\eta_{mp}} \quad \text{.....(III-7)}$$

b) pour une station avec stockage :

Dans ce cas, il y a une partie de l'énergie consommée directement tandis que le reste est stocké puis restitué ce qui entraîne une perte d'énergie supplémentaire due à ce processus.

$$P_{pv} = Q \frac{\rho.g.H_{mt}}{.\eta_{conv}.\eta_{mp}} \left(\frac{T_{ens}}{24\eta_{pv}} + \frac{24 - T_{ens}}{24\eta_{pv}.\eta_{bat}} \right) \quad \text{.....(III-8)}$$

énergie reçue directement énergie emmagasinée

III.5. Estimation du nombre des modules photovoltaïques nécessaires

Le nombre des modules se déduit de :

$$N_{panneau} = \frac{P_{pv}}{P_u} = \frac{P_{pv}}{E.S} \quad \text{.....(III-9)}$$

où

P_u : puissance journalière moyenne par module

E : éclairement journalier moyen sur le site

S : surface efficace de chaque module du panneau (compte tenu de l'orientation)

III.6. Considération du coût

On fait une optimisation sur le coût :

$$c = N_{panneaux} .(c_u + c_{terrain}.S + c_{trans} .M_u) \quad \text{(III-10)}$$

Où :

$N_{panneaux}$: nombre de module

c_u : prix unitaire des modules

$c_{terrain}$: prix du terrain

S : surface du module photovoltaïque

c_{trans} : coût de transport par Kg

M_u : poids de chaque module

Pour chaque type de panneaux solaires, on calcule le coût total et on choisit celui qui demande moins. Et dans certains cas, il faut tenir compte aussi des temps de livraison et de montage des panneaux (par exemple des modules provenant du Japon ou du Sud Est asiatique retardent plus que ceux qui viennent d'Europe, et d'un autre coté, certains panneaux demandent des bâtis spéciaux ou des aménagements particuliers du terrain).

III.7. Dimensionnement des batteries

Nous avons la puissance journalière moyenne qu'il faut stocker dans les batteries :

$$P_{batmoy} = P'_{mp} \frac{24 - T_{ens}}{24} . \frac{1}{\eta_{pv}\eta_{bat}} \quad \text{(III-11)}$$

Cette puissance doit être fournie pendant la durée d'ensoleillement T_{ens} ce qui donne une énergie emmagasinée de :

$$W_{bat} = P'_{mp} \frac{24 - T_{ens}}{24} \cdot \frac{T_{ens}}{\eta_{pv}\eta_{bat}} \quad \text{.....(III-12)}$$

En termes de capacité des batteries (exprimée en Ampères-heures) :

$$C_{tot} = \frac{1}{V_{bat}} P'_{mp} \frac{24 - T_{ens}}{24} \cdot \frac{T_{ens}}{\eta_{pv}\eta_{bat}} \quad \text{.....(III-13)}$$

Où V_{bat} est la tension moyenne de chargement de la batterie.

Le nombre de batteries est donc : $$N_{bat} = \frac{C_{tot}}{C_{bat}} \quad \text{.....(III-14)}$$

Dans ce cas aussi, on peut faire une optimisation sur le coût, mais il faut aussi faire intervenir la durée de vie des batteries.

III.8. Observation (contrôle) d'une station de pompage

Comme dans tous les problèmes de l'automatisation, nous devons d'un coté acquérir les variables du système (notre station) ce qui revient à faire des mesures, et de l'autre coté, donner des commandes ou consignes en fonction des variations dans le système.

On peut envisager plusieurs schémas pour la surveillance de la station selon son type :

On peut ajouter ou supprimer certaines commandes ou certaines mesures selon les disponibilités du système et selon le prix de revient de la station aussi.

III.8.1 Les variables à mesurer

A) Pour le panneau

1- La température ambiante (elle servira pour la mesure de la température de jonction)
2- La tension
3- Le courant débité
4- La direction en sit et/ou azimut si le panneau est orientable

B) Pour le convertisseur

1- La tension de sortie
2- Le courant de sortie
3- La température (pour pouvoir le protéger contre les échauffements excessifs)

C) Pour la batterie

1- Le courant avec son sens

2- La tension

3- L'état de la solution (solution d'acide) : densité, pH,... etc.

D) Pour la motopompe et le système hydraulique

1- Débit du liquide

2- Hauteur du liquide dans le réservoir (s'il y a)et dans la source de puisement (puits ou canalisation...)

3- Pression du liquide au niveau de la pompe

4- Vitesse de rotation du moteur

III.8.2 Les commandes à prévoir

A) Pour le panneau

1- Rotation en site et en azimut

2- Connexion/déconnexion au reste de l'installation

3- Déclenchement d'un éventuel système de refroidissement et/ou nettoyage...

B) Pour le convertisseur

1- Commande des commutations internes (éventuellement) ou leur fréquence

2- Commande de la fréquence (si la $2^{ème}$ commande est impossible)

C) Pour la batterie

1- Connexion/déconnexion en cas d'atteinte de la charge complète pendant le jour

D) Pour la motopompe

1- Connexion/déconnexion en cas de danger quelconque

E) Pour le système hydraulique

1- Commande des vannes pour contrôler le débit

2- Commande des vannes qui contrôlent la direction du liquide s'il doit être distribué sur plusieurs destination

III.8.3 Méthode de la commande

Nous n'allons pas envisager dans ce projet d'utiliser les techniques pointues de la commande. Nous allons nous contenter de réactions simples aux événements qui se produisent dans la station de pompage.

Avant tout nous devons disposer d'un organe d'entrée/sortie adéquat. Nous avons une petite image sur le système qu'il faut :

L'acquisition des données se fait sur place avec des capteurs convenables (diviseurs de tension et shunts pour les grandeurs électriques, des doublets thermiques avec amplificateur, des sondes piézo-électriques des pH-mètres... etc.).

On peut soit munir chaque capteur avec un convertisseur Analogique/Numérique sur place. Soit utiliser un convertisseur Analogique/Numérique multiplexé à un ensemble de capteurs liés à une seule composante.

La transmission entre les convertisseurs Analogique/Numérique et l'ordinateur se fait après multiplexage des entré et sorties des convertisseurs (pour la sélection d'un multiplexeur à la fois) ; puis de sérialiser les données (interface série) ; et enfin, modulation éventuelle et transmission vers l'ordinateur où les données son acquises avec l'interface série ou le modem.

Pour les commandes, l'ordinateur doit être muni d'une carte d'acquisition digitale (DAQ) les commandes sont envoyées vers des organes de commande digitale (moteurs pas à pas pour orienter les panneaux et relais pour brancher/débrancher les différentes composantes).

Remarque : *on peut se passer des interfaces série et de la modulation pour l'acquisition et utiliser la carte DAQ dans le cas des faibles distances entre la station et l'ordinateur (quelques dizaines de mètres).*

CHAPITRE IV

PROGRAMME DE SIMULATION

IV. Description du programme réalisé

Le but essentiel du projet étant de réaliser un programme en Visual Basic pour le monitoring d'une station de pompage, il convient avant d'aborder les programmes de simulation eux-mêmes de donner une idée générale sur ce programme, quelles techniques de programmation est-ce qu'il fait intervenir et quels sont ses éléments constitutifs.

Tout d'abord, un programme en Visual Basic doit tourner sur la plate-forme Windows. Alors nous devons d'abord présenter en quelques lignes les caractéristiques de celle-ci.

IV.1. Principes de la programmation sous Windows-32Bits

La plate-forme Win-32 est une plate-forme, avant tout, graphique très interactive. Les médias qui lui sont absolument nécessaires sont un écran «graphique », une souris et une capacité de stockage énorme (ce point est important pour voir que les exigences de la simulation sont très faibles devant ceux de la présentation).

Du point de vue de l'apparence, elle se présente sous une forme de fenêtres (chaque application doit présenter, au moins une fenêtre) qui se présentent ensembles sur l'écran (c'est le principe du multitâche qui est l'aspect technique de la plate-forme). Chaque fenêtre possède certains éléments standards qui sont communs à toutes les fenêtres. Ce sont la barre du titre avec la case système (la petite icône à droite de cette barre) et les cases d'agrandissement, de rétablissement de la taille et de minimisation), la barre des menus, éventuellement, et les contrôles standards (boutons, boites de saisie, boutons radio, cases à cocher... etc.)

Ces éléments permettent l'interactivité avec l'utilisateur dans le cadre du nécessaire, mais dans le cas où les éléments standards offerts par la plate-forme ne satisferaient pas les besoins de l'interactivité, le programmateur est invité à en créer les siens.

Du point de vue technique, la plate-forme Win-32 possède les caractéristiques suivantes :

- *Obligation de respecter les règles d'ergonomie :*

La présentation graphique de Windows 95 ou NT impose au programmateur de présenter une interface à la fois élégante et opérationnelle et met à sa disposition tous les ingrédients nécessaires pour donner finalement une application facile à apprendre et à utiliser

(l'utilisateur n'a pas besoin de s'initier à l'interface) tout en restant performante et très complexe dans son fonctionnement interne ; sauf s'il veut faire autrement, ce qui est déconseillé pour des applications professionnelles car une norme est prescrite par IBM (avec Graphic Interface Standards) et Microsoft (sous forme de notes pour les programmateurs du SDK ou IDK) qui exige des certaines règles d'ergonomie dont la plus importante est l'uniformité de l'interface avec l'utilisateur (c'est à dire que l'utilisateur doit toujours trouver les mêmes éléments visuels, ainsi que les mêmes actions à entreprendre dans les différentes situations comme les cases d'agrandissement, réduction ou redimensionnement des fenêtres ou la possibilité d'annulation des actions qui peuvent l'être... etc.)

- *Elle est orientée objets :*

C'est à dire qu'elle fait intervenir les principes de la POO (Programmation Orientée Objet). Et pour ceci, elle exige des connaissances dans ce domaine. Les objets de win-32 sont très divers allant des types de structures (on dit classes d'objets en POO) gérées par le système d'exploitation aux entités définies par le programmateur et l'utilisateur.

- *Elle est gérée par des événements :*

Un événement est un incident externe qui intervient de façon inattendue à un moment donné. Parmi ceux-ci, on cite les commandes de l'utilisateur entrées par le clavier ou avec la souris, les impulsions de l'horloge du PC et les données provenant des différents périphériques et des interfaces de communication. Il y en a d'autres définis par le système d'exploitation et par les programmateurs des applications, ils sont plus nombreux que les premiers.

Les événements demandent des traitements appropriés qui doivent être implémentés par le programmateur selon les besoins de l'application. Il suffit ensuite de déclarer ces traitements au système qui s'en occupe (le traitement linéaire ou algorithmique utilisé sous les anciens systèmes d'exploitation ne peut pas parvenir à des résultats aussi performants que ceux des systèmes gérés par les événements comme win-32).

– *Elle nécessite un traitement en temps réel :*

Les événements sous win-32 se comptent en milliers par seconde et le système ne peut en ignorer que très peu. Que faut-il faire si le traitement d'un événement dure pour une longue période ?

Dans un tel cas d'autres événements continuent à parvenir au système qui les empile en vue de les traiter ultérieurement ; mais au bout d'un certain temps, la pile se sature et le système qui arrive à ne rien faire se bloque tout bêtement.

Pour empêcher cette situation, le seul remède est de faire les traitements en temps réel. C'est à dire la réduction des temps de calcul pour tenir en un temps suffisamment faible pour laisser le système libre pour les autres fonctions. Ceci par l'amélioration des algorithmes et des méthodes de réaction des applications (par exemple le découpage d'un long traitement en tranches si on n'arrive pas à le réduire suffisamment).

Ceci pour le système d'exploitation, et pour le compilateur Visual Basic, on va passer en revue quelques-unes unes de ses fonctionnalités et caractéristiques.

IV.2. Principes de programmation avec Visual Basic

Visual Basic est un langage de programmation structuré et orienté objet.

Contrairement à ce que son nom peut évoquer dans certains esprits, Visual Basic n'est pas en aucun cas un langage destiné aux débutants de l'informatique, il possède un «véritable » compilateur et permet la création très aisément de toutes les applications d'envergure moyenne avec des fichiers exécutables et des bibliothèques DLL ainsi que les contrôles ActiveX qui font des pages Web de véritables stations multimédia, ceux-ci étant très difficilement réalisables sous Visual C ou Delphi, les concurrents directs de Visual Basic. En plus de cela, il intègre le puissant moteur de bases de données Jet de Microsoft.

Visual Basic possède un environnement de développement intégré (EDI) qui permet une conception visuelle du programme (c'est à dire l'aspect et contenu des fenêtres).

Avant tout, le programmeur doit concevoir les fenêtres en s'aidant d'une boite à outils riche avec les éléments visuels de Windows (Boutons, Listes, Cases à cocher, Onglets ...) auxquels, il peut ajouter ses propres contrôles ActiveX. Une fenêtre «propriétés » permet de personnaliser ces objets (couleur, état actif ou non, visibilité à l'écran, nom interne, texte ou liste affichés ...).

On note que la propriété la plus importante d'un objet est son nom «name »car c'est avec elle qu'on peut référencer cet objet partout dans le programme.

Après la conception de l'interface graphique, le programmeur doit définir les actions à attribuer aux interactions de l'utilisateur avec ces objets. Expliquons par un exemple :

Un bouton «OK » dans une boite de dialogue après avoir sauvé les données qu'elle contient et qui correspondent aux «choix » de l'utilisateur dans des variables du programme, dans une base de données ou dans les propriétés d'autres objets dans le programme.

Ceci se fait par la programmation des événements attribués à l'objet dans la fenêtre du «code » en choisissant dans les listes du haut de la fenêtre (voir figure IV-1) l'objet et l'événement ; alors, Visual Basic affiche un sous programme relatif à cet événement (voir figure IV-1).

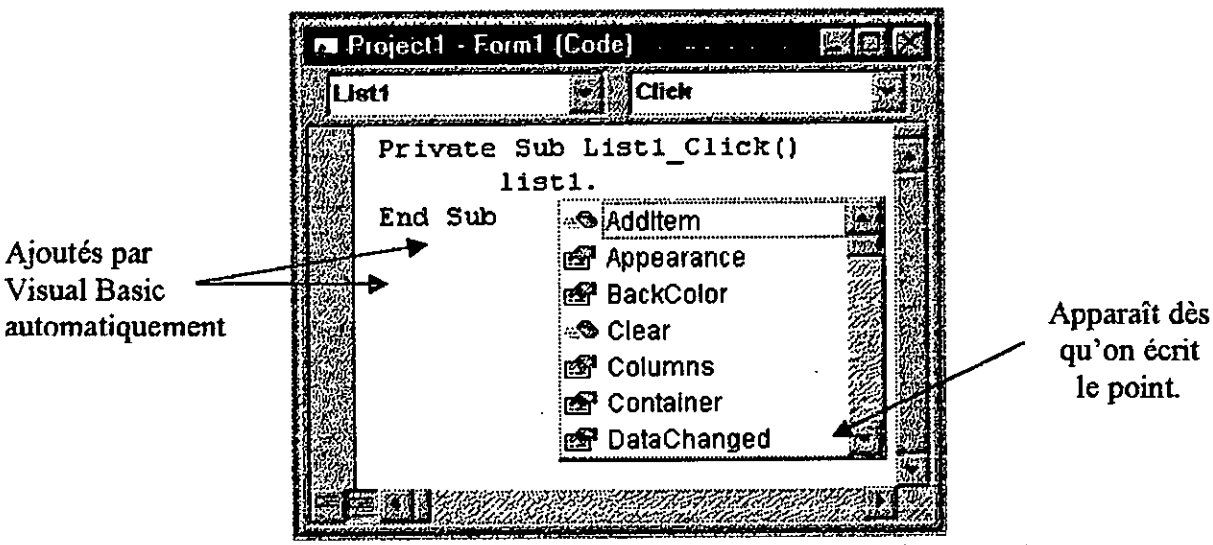


Figure IV-1 : la fenêtre code relative à la fenêtre nommée «Form1 ».

L'intérêt de la programmation sous Visual Basic requiert deux intérêts principaux :

1. Un langage facile qui représente une extension au BASIC connu et ainsi, un débutant qui connaît le BASIC va trouver certainement beaucoup des éléments de syntaxe qu'il connaît déjà ; et surtout, quelqu'un qui maîtrise Quick-BASIC (fourni par Microsoft avec MS-DOS 6) ne trouvera certainement presque aucune différence de syntaxe, mis à part une légère différence dans le point de vue logique de la programmation événementielle inhérent à Windows.
2. L'EDI de Visual Basic dégage le programmeur de la recherche fastidieuse dans les propriétés des objets ou les arguments des sous-programmes ou fonctions puisqu'il affiche des bulles contenant ces informations (voir figure IV-1) dès que le programmeur tape le

nom d'un objet existant suivi d'un point (ce qui veut dire en POO qu'il va introduire une méthode ou une propriété de l'objet) ce qui lui permet choisir la propriété ou méthode qu'il veut dans une liste ; ou bien, lorsqu'il tape le nom d'un sous-programme SUB ou FUNCTION et alors une info bulle lui rappelle ses arguments (ou paramètres). Cette fonctionnalité facilite beaucoup la programmation, améliore la vitesse et facilite aussi l'apprentissage du langage pour les débutants.

Remarque :

Dans Visual Basic, il n'y a pas de différenciation entre majuscules et minuscules, l'éditeur, d'ailleurs, remet les mots tapés en forme comme ils ont étés tapés la première fois ; et s'il s'agit de fonction, d'objet ou de variable déclarée, il les remet comme ils sont écrits dans la déclaration. En plus, il colorie les mots selon leurs catégories syntaxiques (mots réservés, identificateurs d'objets, variables et sous- programmes, commentaires ... etc.).

IV.3. Description du programme réalisé

Le programme présente une fenêtre principale dotée de menus et d'images figurant les parties qui composent la station à simuler, contrôler ou dimensionner. Un clic avec la souris sur l'un des éléments nous conduit au tableau de bord qui lui est destiné.

Le type de station figuré au lancement du programme est une station très simple composée par un panneau solaire et la motopompe (moteur à courant continu) avec un puits et un réservoir.

Selon la volonté de l'utilisateur, il peut ajouter un convertisseur électrique (onduleur) et/ou batteries de stockage et ceci à travers le menu «station ».

IV.1.1. Les menus

A) Menu «Quitter » :

Il permet de faire fin au programme au cas ou l'utilisateur ne sait pas ou ne veut pas utiliser la case de fermeture () ou la case système ().

B) Menu «Station » :

Il permet de déterminer le type de station en déterminant l'existence ou non de certains éléments de la stations. Ceci est immédiatement illustré sur l'écran par l'apparition ou la disparition de ces éléments. Il contient deux options :

1. Sous-menu "Convertisseur" :

Il permet d'ajouter ou de supprimer le convertisseur selon que le moteur est à courant continu (supprimé) ou à courant alternatif (ajouté).

2. Sous-menu "Batterie" :

Il permet d'ajouter ou de supprimer la batterie de stockage.

C) Menu «Propriétés » :

Il permet le choix manuel des types des composantes de la station (dans le cas où le dimensionnement serait déjà fait ou que la station existe déjà, le cas de la surveiller), il contient 4 sous-menus :

1. Sous-menu "Panneau" :

Il permet d'accéder à une boite de dialogue "Propriétés du panneau" qui permet le choix du constructeur et du type du panneau ; et aussi de déterminer s'il est ou non orientable et dans quelle direction (site et/ou azimut).

Elle donne finalement un résumé sur les propriétés du module choisi (puissance maximale délivrée lorsque le panneau est éclairé avec 800 W/m², tension de circuit ouvert, courant de court-circuit, surface, prix unitaire ... etc.)

Cette boite de dialogue permet aussi à l'utilisateur d'enregistrer de nouveaux constructeurs et de nouveaux types de panneaux dont il aurait trouvé les caractéristiques.

2. Sous-menu "Batterie" :

Il permet d'accéder à une boite de dialogue "Propriétés de la batterie" qui permet le choix du constructeur et du type du de la batterie ; et donne un aperçu sur ses propriétés (capacité maximale de stockage, capacité résiduelle minimale admissible, tension de décharge et celle de la charge maximale, le rendement énergétique, le matériau de fabrication et si elle est scellée ou non, poids, prix unitaire ... etc.)

Cette boite de dialogue permet aussi à l'utilisateur d'enregistrer de nouveaux constructeurs et de nouveaux types de batteries dont il aurait trouvé les caractéristiques.

3. Sous-menu "Convertisseur" :

Comme il n'y a pas sur le marché assez d'onduleur qui satisfont les exigences du photovoltaïque (rendement très élevé sur un large éventail de puissances), alors, il est préférable de lui faire une conception sur mesure et de le construire soi-même.

Avec ces contraintes, la boite de dialogue "Propriétés du convertisseur" permet uniquement de choisir le rendement et coût du convertisseur (ceci surtout dans le cadre de ce projet mais une continuation multidisciplinaire surtout avec des électrotechniciens, on peut ajouter d'autres fonctionnalités à cette boite de dialogue).

Remarque : *on peut envisager d'afficher une courbe du rendement en fonction de la puissance, de la munir de prises que l'utilisateur peut déplacer pour obtenir la courbe qu'il a ou qu'il veut réaliser.*

4. Sous-menu "Motopompe" :

Il permet d'accéder à une boite de dialogue "Propriétés de la motopompe" avec laquelle, l'utilisateur choisit le constructeur et le type de la motopompe.

Un autre onglet permet le choix entre pompe émergée ou immergée.

Un troisième onglet donne un aperçu sur ses propriétés (moteur à courant continu ou alternatif, la puissance nominale ... etc.)

Cette boite de dialogue permet également à l'utilisateur d'enregistrer de nouveaux constructeurs et de nouveaux types de motopompes dont il aurait trouvé les caractéristiques.

D) Menu «Activer » :

C'est le moyen d'activer les tableaux de contrôle avec le clavier (conformément à la règle d'ergonomie qui dit de rendre nombreux les moyens d'accès aux fonctionnalités du logiciel en permettant aux gens qui utilisent mal ou qui n'ont pas la souris d'y arriver).

Il contient évidemment les sous-menus suivants :

1. Sous-munu "Panneau"
2. Sous-munu "Batterie"
3. Sous-munu "Convertisseur"
4. Sous-munu "Motopompe"
5. Sous-munu "Puits"
6. Sous-munu "Réservoir"

IV.1.2. Les tableaux de contrôle

Ils se présentent sous forme de fenêtres indépendantes et d'apparences différentes.

A) Fenêtre panneau

Elle permet la surveillance du panneau manuellement en affichant :

- Des étiquettes affichant l'état mesuré du panneau : tension, courant, angles d'orientation en site et en azimut ainsi que la température.

  Si la carte d'acquisition est absente, alors ces étiquettes sont désactivées.

- Le type choisi par l'utilisateur dans "Propriétés du panneau" (le logiciel ne peut pas de lui-même détecter le type, mais il peut vérifier si un type choisi correspond aux données tension et courant acquises par la carte d'acquisition DAQ).

- Un bouton permettant d'afficher la fenêtre de simulation de la position du soleil et de l'ensoleillement.

- Un autre bouton permettant d'afficher la fenêtre de la courbe simulation.

- Un bouton pour brancher ou débrancher le panneau (le mettre hors circuit).

- Des curseurs (Sliders) permettant d'actionner manuellement l'orientation du panneau. Un choix par boutons radio permet de passer en mode automatique.

- Et enfin, un bouton pour afficher l'historique de l'acquisition (fenêtre "Historique ").

Les fenêtres liées à cette fenêtre :

1. Fenêtre de simulation de l'ensoleillement

Elle affiche l'ensoleillement reçu par le panneau à tout moment de n'importe quel jour de l'année et selon l'orienatabilité du panneau. Elle affiche aussi sur un système d'axes tridimensionnel (Sud, Est, Verticale) la position du soleil à ce moment avec des étiquettes qui donnent en chiffres le site et l'azimut de cette position.

Au début, la fenêtre affiche le moment actuel, mais l'utilisateur peut changer le moment et le jour à sa volonté.

2. Fenêtre de la courbe de simulation

Dans cette fenêtre, on peut observer la courbe I = F (V) d'un panneau selon le mode Singer.

Les données (V_{co}, I_{cc}, P_m) sont données par le type de panneau choisi (dans la boite de dialogue "Propriétés du panneau") et l'utilisateur peut comme même les modifier pour voir le comportement du panneau si ces données changent, par exemple.

La courbe se dessine dans une grille munie d'axes gradués. L'utilisateur peut en appuyant sur le bouton de la souris à l'intérieur de la grille de voir un curseur qui lui permet de repérer, avec précision, tout point de la courbe (inspiré du curseur de l'oscilloscope de PSPICE).

3. Fenêtre de l'historique

Elle affiche les mesures effectuées par la carte d'acquisition digitale (DAQ) ; c'est à dire tension, courant et température pendant un certain laps de temps déterminé par N échantillons (ce nombre est choisi dans la même fenêtre) précédants.

L'affichage se fait sous forme de courbes. Ainsi cette fenêtre permet la vérification, la comparaison et l'archivage.

Cette fenêtre n'est pas disponible en l'absence de carte d'acquisition.

B) Fenêtre "Batterie"

Elle permet de surveiller la batterie. Elle affiche :

- Des étiquettes pour la tension, le sens et la valeur du courant et éventuellement, la concentration de l'acide mesuré par la carte d'acquisition.
- Une étiquette illustrée graphiquement pour la charge actuelle de la batterie (en A.h) vis à vis du minimum et maximum admissibles.
- Un bouton pour accéder à la boite de dialogue "Propriétés de la batterie".
- Des étiquettes pour rappeler le type et les caractéristiques de la batterie choisie dans la boite de dialogue "Propriétés de la batterie".
- Un bouton qui permet de passer entre le contrôle manuel et le contrôle automatique.
- Un bouton qui permet de débrancher la batterie quand elle est trop chargée et de la brancher quand le panneau ne fournit pas assez d'énergie pour la motopompe.

Cette fenêtre n'est pas disponible si le type de station considéré ne contient pas de batteries.

Certaines étiquettes et contrôles son désactivé en l'absence de carte d'acquisition.

C) Fenêtre "Convertisseur"

Elle affiche :

- La tension, courant et fréquence de sortie mesurés.
- Un bouton pour accéder à la boite de dialogue "Propriétés du convertisseur".
- Des étiquettes pour rappeler les caractéristiques du convertisseur choisi dans la boite de dialogue "Propriétés du convertisseur ".
- Des graphismes montrant visuellement les commutations dans le convertisseur.

- Une zone de saisie pour le réglage de la fréquence manuellement.

  Cette fenêtre n'est pas disponible si le type de station considéré ne contient pas de convertisseur (motopompe à courant continu).

  Certaines étiquettes et contrôles son désactivé en l'absence de carte d'acquisition.

D) Fenêtre "Motopompe"

Elle affiche :

- La vitesse de rotation et la pression dans la motopompe.
- Un bouton pour accéder à la boite de dialogue "Propriétés de la motopompe ".
- Des étiquettes pour rappeler les caractéristiques de la motopompe choisie dans la boite de dialogue "Propriétés de la motopompe " : rendement, puissance nominale (ou maximale dans notre cas) ainsi que le constructeur.
- Un bouton pour la mise hors circuit de la motopompe en cas de problème quelconque (haute pression, échauffement fort ... etc.)

E) Fenêtre "Puits et réservoir"

Elle représente tout le système hydraulique. Elle affiche et permet de modifier :

- La hauteur du réservoir par rapport à la pompe.
- Le débit du liquide.
- La hauteur de la pompe par rapport à la surface du liquide (cas de pompe immergée) ou la profondeur de la première par rapport à la dernière (cas de pompe émergée).
- Le diamètre des canalisations et le facteur des pertes α

$$Ppertes = \alpha . Q$$

- Des emplacements pour visualiser et commander l'état d'un certain nombre de vannes selon le choix de l'utilisateur à condition que ce nombre ne dépasse plus 8.

IV.1.3. La fenêtre de dimensionnement

Elle passe avec l'utilisateur par toutes les étapes nécessaires selon un ensemble de boutons à la manière des programmes d'installation :

- Bouton "suivant >",
- Bouton "< précédant",
- Bouton "Terminer",
- Et un bouton d'annulation.

A l'appui sur le bouton "suivant >", si l'utilisateur a entré une valeur inacceptable ou s'il a oublié une donnée obligatoire, il est averti. Certains champs ne doivent pas rester blancs ; alors, ils sont activés les premiers.

Les étapes sont :

A) Etape de détermination des variables d'entrée

Ces variables sont :

- Le débit Q.
- Le nombre d'heures de pompage pour calculer le débit moyen journalier (si la station est à stockage et fonctionne de façon ininterrompue on donne 24 heures).
- La hauteur du réservoir H_1 et celle de la pompe à la surface du liquide dans le puits (si la pompe est émergée alors la valeur est négative).
- Le facteur de pertes α.

Le programme affiche initialement les valeurs contenues dans les autres fenêtres pour faciliter la tâche de l'utilisateur mais celui-ci peut les changer à sa volonté.

B) Etape de détermination des facteurs économiques pouvant intervenir.

- Le prix unitaire des panneaux et des batteries ainsi que celui de la motopompe.
- Le coût de transport des panneaux (par kg ou par panneau) et celui des batteries (par unité).
- Le coût des études relatives à la conception du convertisseur ou bien son prix d'achat + transport.

C) Etape de choix du critère de l'optimisation

L'utilisateur a le choix de l'optimisation selon :

- Le coût total,
- Le nombre d'unités,
- La surface occupée.

D) Dernière étape : Résultats de dimensionnement

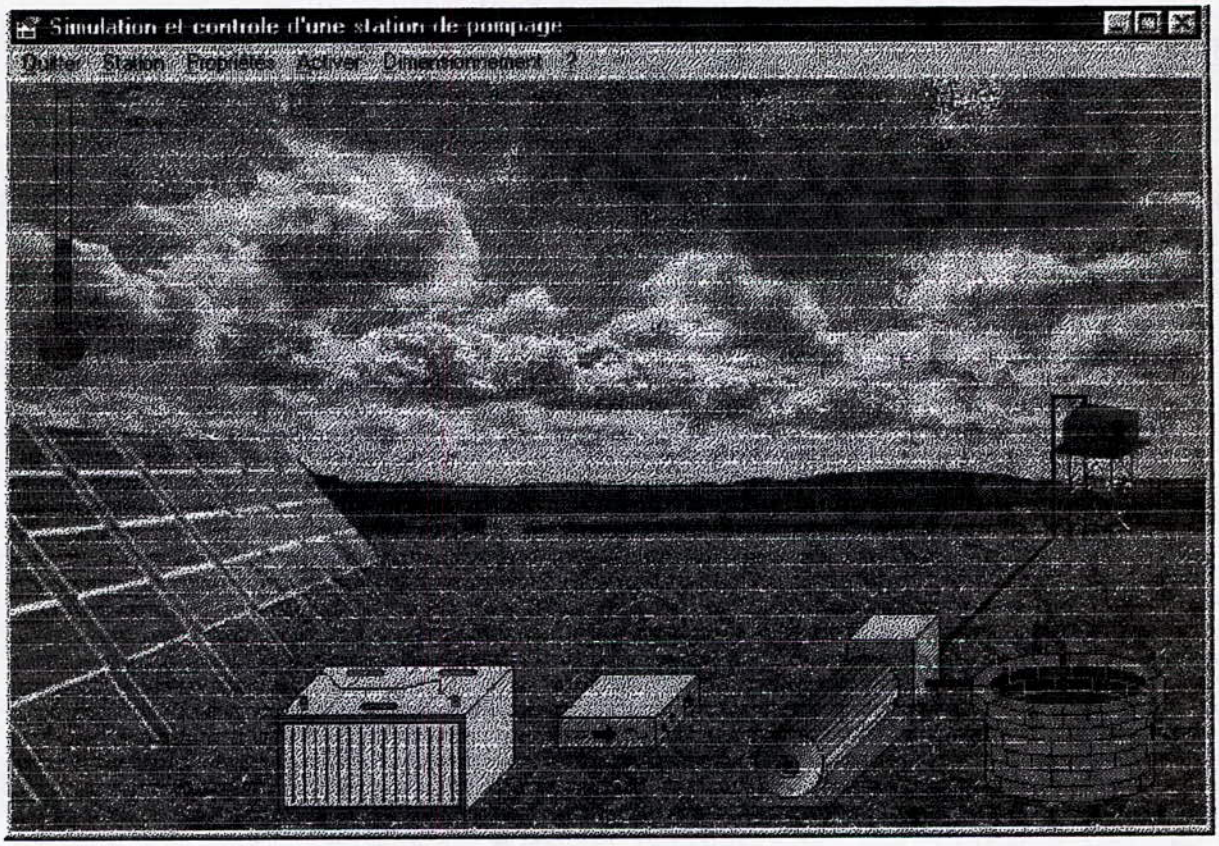
Le programme affiche :

- Le nombre des panneaux,
- Le coût total des panneaux,
- Le nombre des batteries (s'il n'y a pas de batteries dans le type de station considéré, ce nombre est nul),
- Le coût total des batteries,
- Le coût du reste des composants (convertisseur, éventuellement, + motopompe),
- Et enfin, le coût total.

ANNEXE

Dans cette annexe, nous allons donner des figures sur les fenêtres du programme réalisé. Ces illustrations ont été obtenues grâce à l'utilisation de la touche Alt + Impr écran pendant l'exécution du programme.

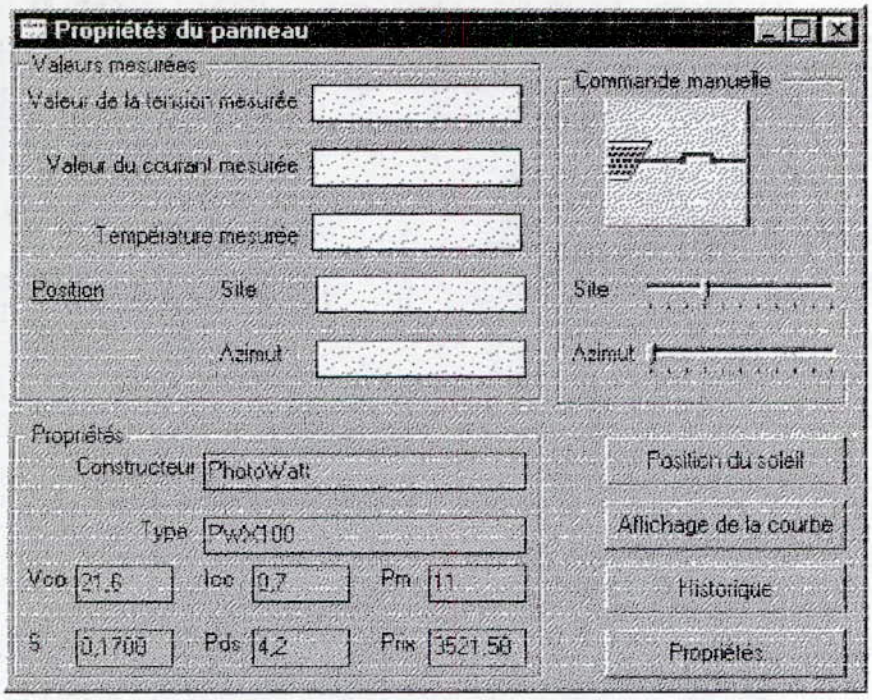
La première figure représente la fenêtre de démarrage du programme (celle qui contient les menus et l'interface globale du programme).



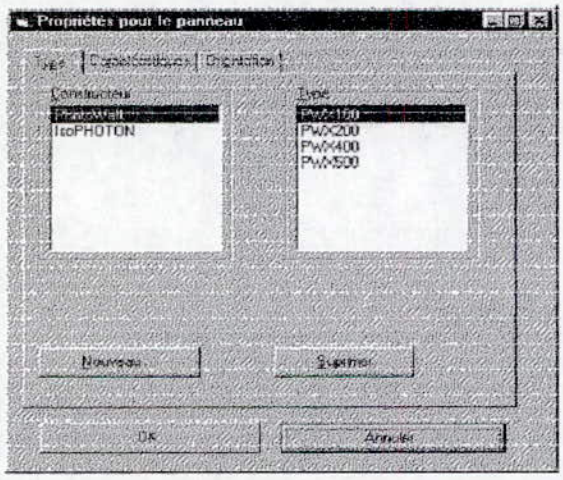
On y remarque les composantes d'une station de pompage photovoltaïque complète. Le thermomètre, en haut à gauche) permet (pour la version actuelle) de donner la température ambiante. Il permet aussi, si une carte DAQ est branchée d'afficher la valeur mesurée.

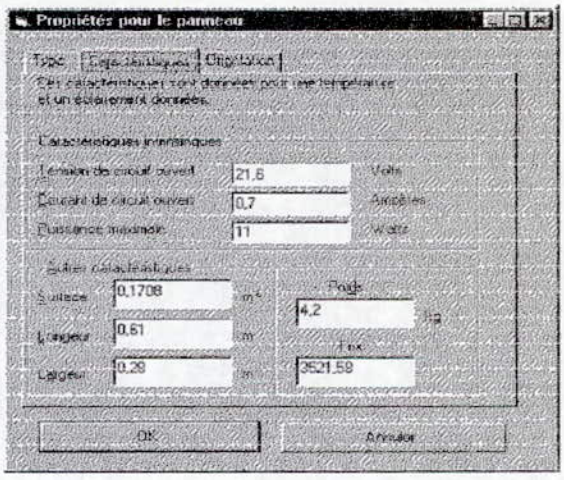
Dans ce qui suit, nous donnons les fenêtres des différents tableaux de bord :

Le tableau de bord du panneau :

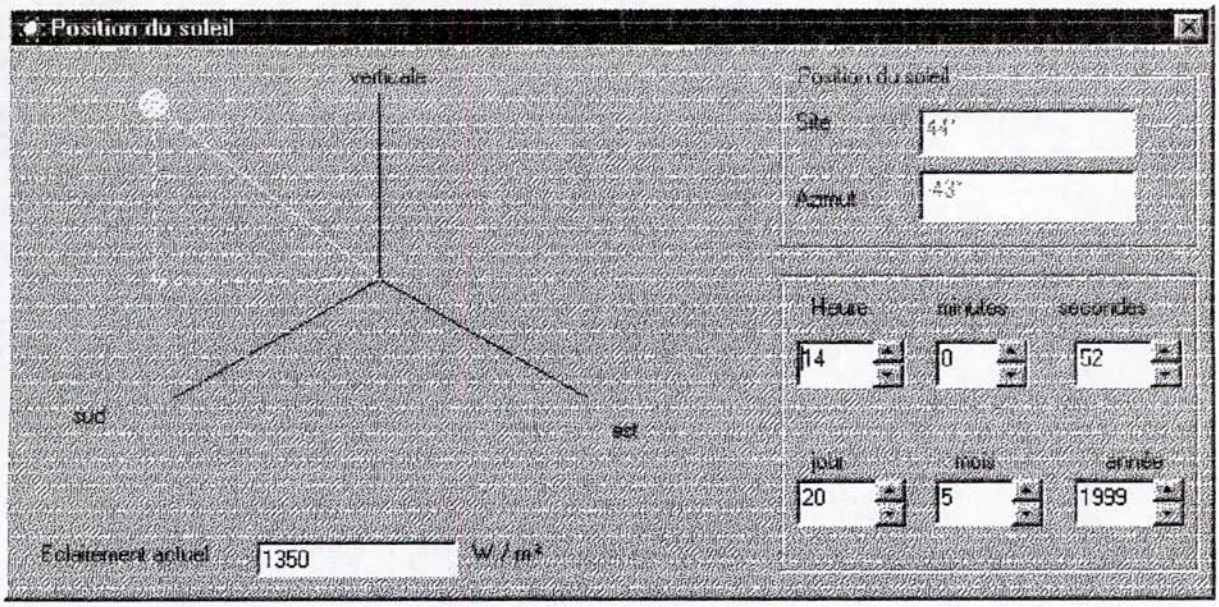


La fenêtre «propriétés du panneau » :

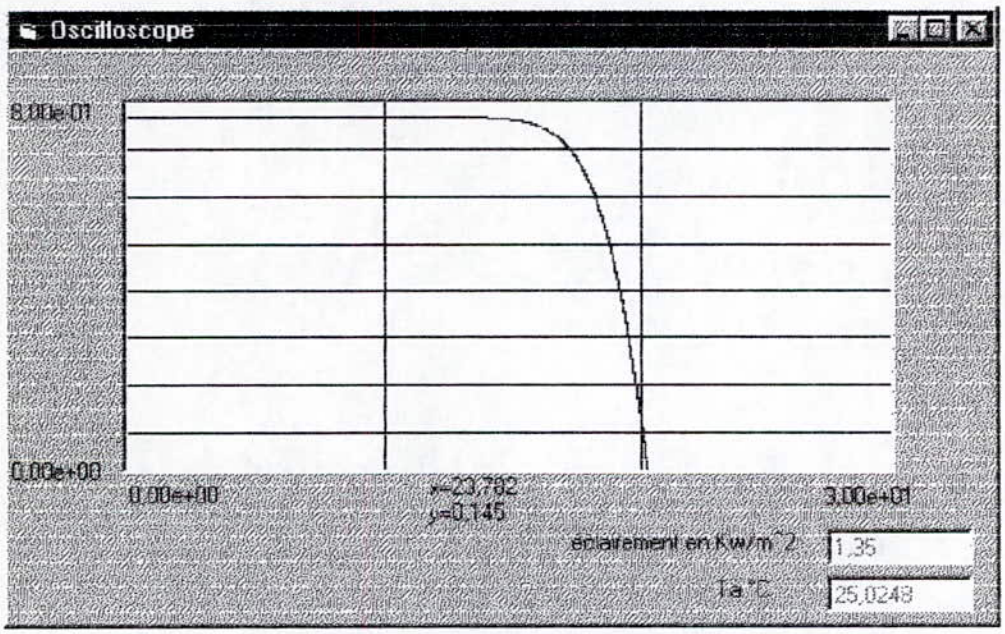




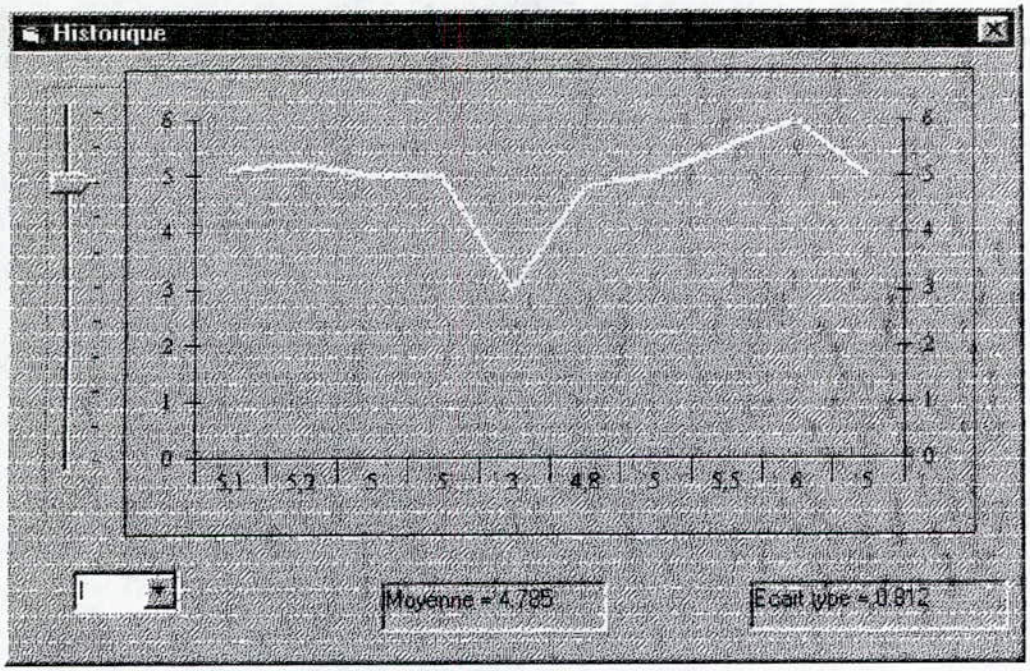
La fenêtre de la position du soleil et de l'ensoleillement :



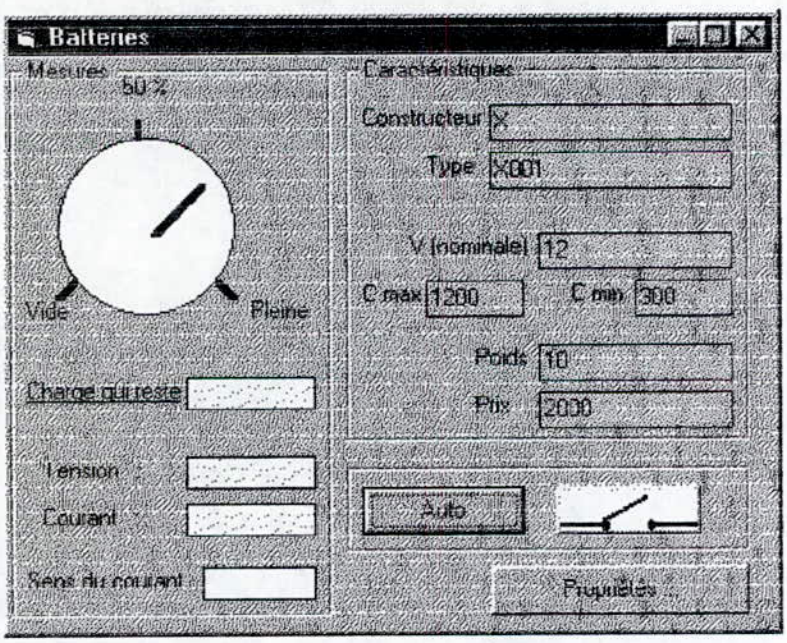
La fenêtre d'affichage de la courbe :



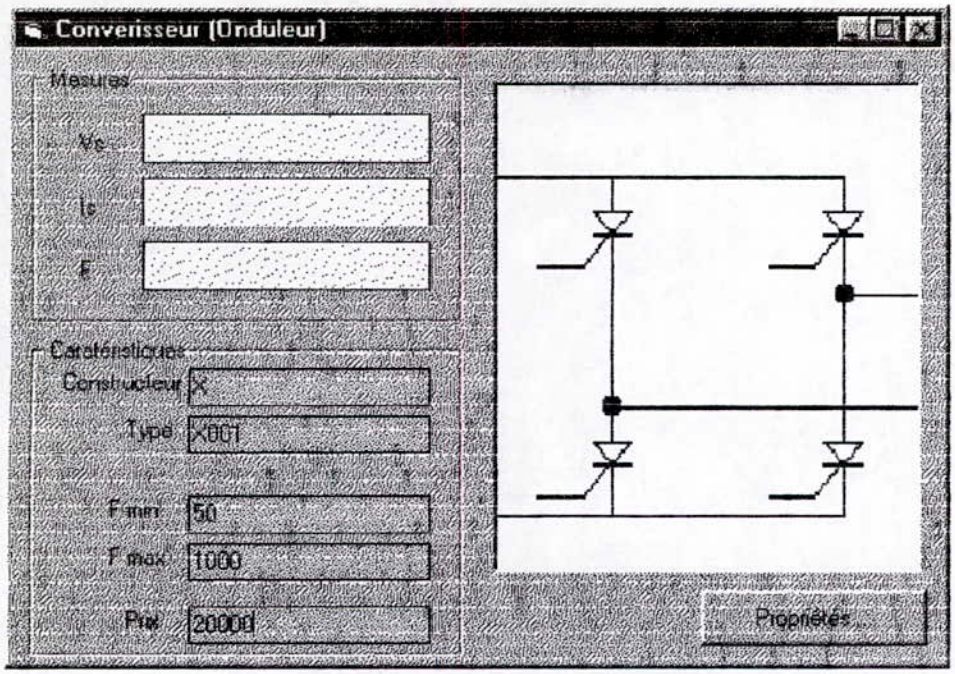
La fenêtre de l'historique :



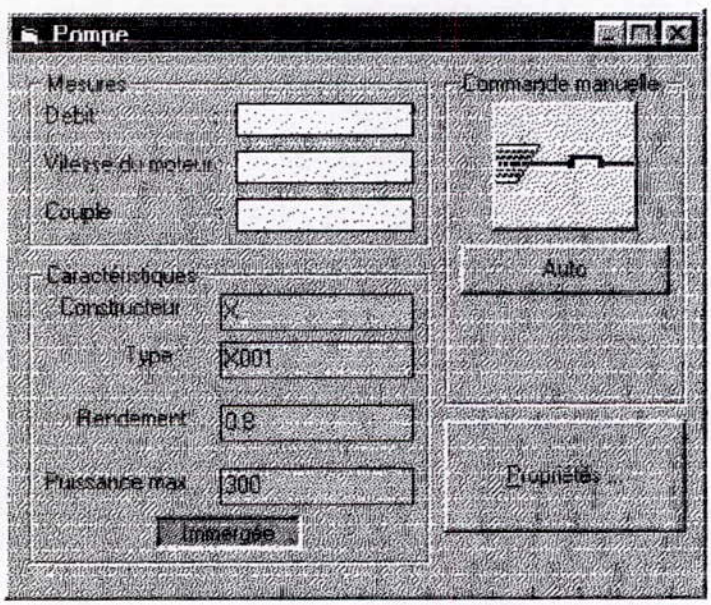
Le tableau de bord de la batterie :



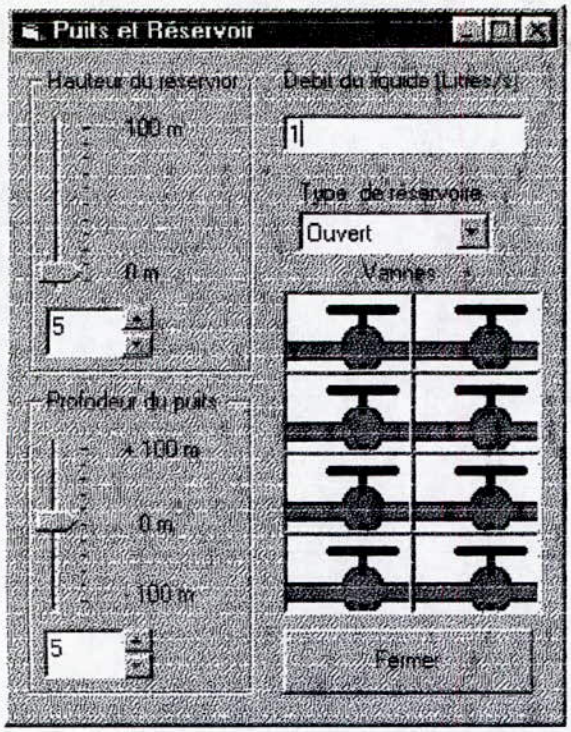
Le tableau de bord du convertisseur :



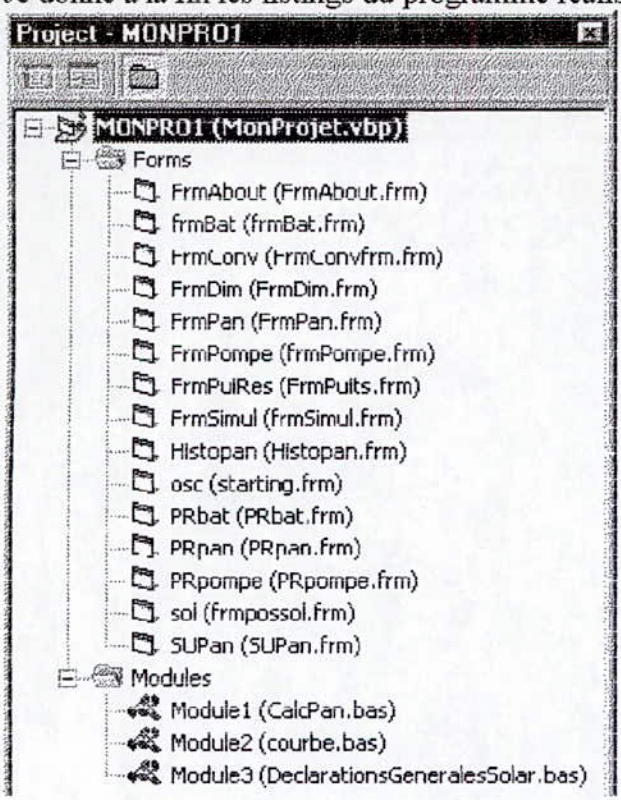
Le tableau de bord de la pompe :



Le tableau de bord du système hydraulique :



Je donne à la fin les listings du programme réalisé ; et ceci après la constitution du projet :



1. *Listing de «module3 » (declaration des variables et fonctions de portée globale) :*

```
Option Explicit
Global Const Pi = 3.1415926

'variables pour la lecture des fichiers de données
Global Const MonChemain As String = "c:\bouhebbal\source\"
Global NoFichier As Integer, ChaineLue As String
'Panneau liste
Global ConstrctPAN() As String, TypesPAN() As String, LstVPan() As Single, LstI() As
Single, LstPPAN() As Single, LstL() As Single, LstW() As Single, LstPdPAN() As Single,
LstPrixPAN() As Single, LstRendPan() As Single
Global NLstPan As Integer, IdxLstPan() As Integer, ndxLstPan As Integer
'pompe liste
Global ConstrctPompe() As String, TypesPompe() As String, LstVPompe() As Single,
LstPPompe() As Single, LstDPompe() As Single, LstPrixPompe() As Single,
LstRendPompe() As Single, LstImmersion() As Single
Global NLstPompe As Integer, IdxLstPompe() As Integer, ndxLstPompe As Integer
'batterie liste
Global ConstrctBat() As String, TypesBat() As String, LstVnBat() As Single, LstVmaxBat()
As Single, LstVminBat() As Single, LstCmaxBat() As Single, LstCminBat() As Single,
LstPrixBat() As Single, LstRendBat() As Single
Global NLstBat As Integer, IdxLstBat() As Integer, ndxLstBat As Integer

'Déclaration des variables globales
Global TExterne As Single, DAQ As Boolean
```

```
'panneau ,pompe, convertisseur et batterie choisis
Global PanActuel As Integer, PompeActuelle As Integer, BatActuelle As Integer, ConvActuel
As Integer

'1)panneau
'Parametres de simulation
Global Const Alpha As Double = 0.0025, Beta As Double = 0.5, Gamma As Double =
0.00288
Global Rs As Double, Lamda As Double
'variables
Global EnSol As Single ' Ensoleillement
Global TPanneau As Single, VPanneau As Single, IPanneau As Single, Ppanneau As Single
Global AngleSite As Single, AngleAzimut As Single
Global Orientation As Integer '0=none, 1=site, 2= azimut, 3= les deux
'Pour le soleil
Global NJours(12) As Long

'2)pompe
Global Debit As Single, Heff As Single
Global TMoteur As Single, VitMoteur As Single

'3)convertisseur
Global VConv As Single, IConv As Single, FConv As Single

'4)batterie
Global VBatterie As Single, IBatterie As Single
'5)systeme hydraulique
Global H1 As Single, H2 As Single

'fonctions d'ordre global
Sub InitGlobal()
 Call InitSol
 Call LirePanneaux
 Call LirePompe
 Call LireBat

' IPhoto = 0.03!
' A = 2
' KT_sur_e = 0.026 * A
' Isat = 0.0000001
' Rcp = 100!
' Ecp = 0!
 Call changeTExterne(25)
' Call calculensol
 VPanneau = 0.6
 IPanneau = 0.01
' Call changeTPanneau(TExterne + 30 * EnSol)
End Sub
```

```
Sub InitSol()
 NJours(1) = 0
 NJours(2) = 31
 NJours(3) = 28 + NJours(2)
 NJours(4) = 31 + NJours(3)
 NJours(5) = 30 + NJours(4)
 NJours(6) = 31 + NJours(5)
 NJours(7) = 30 + NJours(6)
 NJours(8) = 31 + NJours(7)
 NJours(9) = 31 + NJours(8)
 NJours(10) = 30 + NJours(9)
 NJours(11) = 31 + NJours(10)
 NJours(12) = 30 + NJours(11)
 EnSol = 1.35
End Sub


Sub LirePanneaux()
 NoFichier = FreeFile()
 NLstPan = 0
 ndxLstPan = 0
 Open MonChemain + "mesdonnéesPan.dat" For Input As #NoFichier
 Do While Not EOF(NoFichier)
  Input #NoFichier, ChaineLue
  If Left(ChaineLue, 1) = "&" And Mid(ChaineLue, 2) <> "" Then
   PRpan.List1.AddItem Mid(ChaineLue, 2)
   ReDim Preserve IdxLstPan(ndxLstPan)
   ReDim Preserve ConstrctPAN(ndxLstPan)
   ConstrctPAN(ndxLstPan) = Mid(ChaineLue, 2)
   IdxLstPan(ndxLstPan) = NLstPan
   ndxLstPan = ndxLstPan + 1
  Else
   ReDim Preserve TypesPAN(NLstPan)
   ReDim Preserve LstVPan(NLstPan)
   ReDim Preserve LstI(NLstPan)
   ReDim Preserve LstPPAN(NLstPan)
   ReDim Preserve LstL(NLstPan)
   ReDim Preserve LstW(NLstPan)
   ReDim Preserve LstPdPAN(NLstPan)
   ReDim Preserve LstPrixPAN(NLstPan)
   ReDim Preserve LstRendPan(NLstPan)

   TypesPAN(NLstPan) = ChaineLue
  Input #NoFichier, ChaineLue
   LstVPan(NLstPan) = Val(ChaineLue)
  Input #NoFichier, ChaineLue
   LstI(NLstPan) = Val(ChaineLue)
  Input #NoFichier, ChaineLue
   LstPPAN(NLstPan) = Val(ChaineLue)
  Input #NoFichier, ChaineLue
```

```
  LstL(NLstPan) = Val(ChaineLue)
 Input #NoFichier, ChaineLue
  LstW(NLstPan) = Val(ChaineLue)
 Input #NoFichier, ChaineLue
  LstPdPAN(NLstPan) = Val(ChaineLue)
 Input #NoFichier, ChaineLue
  LstPrixPAN(NLstPan) = Val(ChaineLue)
 Input #NoFichier, ChaineLue
  LstRendPan(NLstPan) = Val(ChaineLue)
  NLstPan = NLstPan + 1
 End If
Loop
ReDim Preserve IdxLstPan(ndxLstPan)
IdxLstPan(ndxLstPan) = NLstPan
PRpan.List1.Selected(0) = True
PRpan.List2.Selected(0) = True
Close (NoFichier)
PanActuel = 0
End Sub

Sub LirePompe()
 NoFichier = FreeFile()
 NLstPompe = 0
 ndxLstPompe = 0
 Open MonChemain + "mesdonnéesPompe.dat" For Input As #NoFichier
 Do While Not EOF(NoFichier)
  Input #NoFichier, ChaineLue
  If Left(ChaineLue, 1) = "&" And Mid(ChaineLue, 2) <> "" Then
   PRpompe.List1.AddItem Mid(ChaineLue, 2)
   ReDim Preserve IdxLstPompe(ndxLstPompe)
   ReDim Preserve ConstrctPompe(ndxLstPompe)
   ConstrctPompe(ndxLstPompe) = Mid(ChaineLue, 2)
   IdxLstPompe(ndxLstPompe) = NLstPompe
   ndxLstPompe = ndxLstPompe + 1
  Else
   ReDim Preserve TypesPompe(NLstPompe)
   ReDim Preserve LstVPompe(NLstPompe)
   ReDim Preserve LstPPompe(NLstPompe)
   ReDim Preserve LstRendPompe(NLstPompe)
   ReDim Preserve LstDPompe(NLstPompe)
   ReDim Preserve LstPrixPompe(NLstPompe)
   ReDim Preserve LstImmersion(NLstPompe)

   TypesPompe(NLstPompe) = ChaineLue

  Input #NoFichier, ChaineLue
   LstVPompe(NLstPompe) = Val(ChaineLue)
  Input #NoFichier, ChaineLue
   LstPPompe(NLstPompe) = Val(ChaineLue)
  Input #NoFichier, ChaineLue
```

```
  LstRendPompe(NLstPompe) = Val(ChaineLue)
  Input #NoFichier, ChaineLue
  LstDPompe(NLstPompe) = Val(ChaineLue)
  Input #NoFichier, ChaineLue
  LstPrixPompe(NLstPompe) = Val(ChaineLue)
  Input #NoFichier, ChaineLue
  LstImmersion(NLstPompe) = Val(ChaineLue)

  NLstPompe = NLstPompe + 1
 End If
 Loop
 ReDim Preserve IdxLstPompe(ndxLstPompe)
 IdxLstPompe(ndxLstPompe) = NLstPompe
 PRpompe.List1.Selected(0) = True
 PRpompe.List2.Selected(0) = True
 Close (NoFichier)
 PompeActuelle = 0
End Sub

Sub LireBat()
 NoFichier = FreeFile()
 NLstBat = 0
 ndxLstBat = 0
 Open MonChemain + "mesdonnéesBat.dat" For Input As #NoFichier
 Do While Not EOF(NoFichier)
  Input #NoFichier, ChaineLue
  If Left(ChaineLue, 1) = "&" And Mid(ChaineLue, 2) <> "" Then
   PRbat.List1.AddItem Mid(ChaineLue, 2)
   ReDim Preserve IdxLstBat(ndxLstBat)
   ReDim Preserve ConstrctBat(ndxLstBat)
   ConstrctBat(ndxLstBat) = Mid(ChaineLue, 2)
   IdxLstBat(ndxLstBat) = NLstBat
   ndxLstBat = ndxLstBat + 1
  Else
   ReDim Preserve TypesBat(NLstBat)
   ReDim Preserve LstVnBat(NLstBat)
   ReDim Preserve LstVminBat(NLstBat)
   ReDim Preserve LstVmaxBat(NLstBat)
   ReDim Preserve LstCminBat(NLstBat)
   ReDim Preserve LstCmaxBat(NLstBat)
   ReDim Preserve LstRendBat(NLstBat)
   ReDim Preserve LstPrixBat(NLstBat)
   TypesBat(NLstBat) = ChaineLue
  Input #NoFichier, ChaineLue
   LstVnBat(NLstBat) = Val(ChaineLue)
  Input #NoFichier, ChaineLue
   LstVminBat(NLstBat) = Val(ChaineLue)
  Input #NoFichier, ChaineLue
   LstVmaxBat(NLstBat) = Val(ChaineLue)
  Input #NoFichier, ChaineLue
```

```
  LstCminBat(NLstBat) = Val(ChaineLue)
 Input #NoFichier, ChaineLue
  LstCmaxBat(NLstBat) = Val(ChaineLue)
 Input #NoFichier, ChaineLue
  LstRendBat(NLstBat) = Val(ChaineLue)
 Input #NoFichier, ChaineLue
  LstPrixBat(NLstBat) = Val(ChaineLue)
  NLstBat = NLstBat + 1
 End If
 Loop
 ReDim Preserve IdxLstBat(ndxLstBat)
 IdxLstBat(ndxLstBat) = NLstBat
 PRbat.List1.Selected(0) = True
 PRbat.List2.Selected(0) = True
 Close (NoFichier)
 BatActuelle = 0
End Sub

Function Constructeur(n As Integer) As String
 Dim I As Integer
 For I = 0 To ndxLstPan
  If PanActuel < IdxLstPan(I) Then Exit For
 Next
 Constructeur = ConstrctPAN(I - 1)
End Function
Sub changeTExterne(ByVal X As Single)
 TExterne = X
 FrmSimul.Label2 = Format(X, "### °C")
 FrmSimul.Shape3.Top = FrmSimul.Shape1.Top + FrmSimul.Shape1.Height * (1! - ((X + 20)
/ 120!))
 FrmSimul.Shape3.Height = ((X + 20) / 120!) * FrmSimul.Shape1.Height
 TPanneau = TExterne + 30 * EnSol
End Sub
'Function IdeV(v As Single) As Single
' IdeV = IPhoto - Isat * (Exp(v / KT_sur_e) - 1!)
'End Function
'Sub Resoudre(I As Single, V As Single)
' Dim k As Integer, d As Single, dv As Single
' V = 1!
' dv = 1!
' d = (IdeV(V + 0.000001!) - IdeV(V - 0.000001!)) / 0.000002! - 1! / Rcp
' Do While (Abs(dv) > 0.000001!)
'   dv = -(IdeV(V) - (V - Ecp) / Rcp) / d
'   V = V + dv
'   I = (V - Ecp) / Rcp
' Loop
'End Sub
'Sub changeTPanneau(ByVal x As Single)
'If TPanneau > 0 Then
' Isat = Isat * (x / TPanneau) ^ 3
```

```
' KT_sur_e = KT_sur_e * (x / TPanneau)
'End If
' TPanneau = x
' FrmPan.Text3 = Format(x, "###.## °K")
' Call Resoudre(IPanneau, VPanneau)
' FrmPan.Text1 = VPanneau
' FrmPan.Text2 = IPanneau
' Call courbe(Module1.x(), Y())
'End Sub
Sub ChangeDebit(ByVal X As Single)
 Debit = X
 FrmPompe.Label4 = Format(X, "###.## [litres/min]")
End Sub
Sub changeHeff(ByVal X As Single)
 Heff = X
 FrmPompe.Label6 = Format(X, "###.## [metres]")
End Sub
```

2. *Listing de «module1 » (calcul des paramètres de simulation du panneau) :*

```
Option Explicit
Const T0 = 25
Private Voc As Double, Icc As Double, Pm As Double, Im As Double
Private Voco As Double, Icco As Double, Pmo As Double
Private indx As Double
Private I As Integer, Fin As Boolean
Function Vdel(ByVal I As Double) As Double
 Vdel = Voc * (1 + Log((Icc - I) / Icc) / 20.7) - Rs * I
End Function
Sub ChercherRs()
 Dim I As Double, p As Double
 Dim dRs As Double, dI As Double
 Rs = 0 ' valeur initiale de rs
 dRs = 1 ' valeur d'incrément
 dI = Icc / 1000 ' résolution en courant
            ' on divise [0, Icc]en 1000 segments
            ' puis on calcule la puissance à chacun.
 GoTo truc
 Do
  Do
   Rs = Rs + dRs ' nouvelle valeur de Rs
   p = 0 'On cherche le max de p à partir de 0
   For I = 0 To Icc Step dI
    If p < I * Vdel(I) Then p = Vdel(I) * I
    ' si max actuel< puissance actuelle alors...
   Next
   DoEvents
   If Fin Then Exit Sub
   ' ici p=max de puissance à cette valeur de Rs
  Loop Until (p < Pm) Or Fin 'on augmente  Rs jusq'à ce que p soit < Pm
```

```
 'Loop Until (p < Pm)
 dRs = -dRs / 10
' changer de sens de déplacement
' avec une précision 10 fois meilleure.
' maintenant on diminue ; on revient en arrière.
 Do
  Rs = Rs + dRs ' nouvelle valeur de Rs
  p = 0 'On cherche le max de p à partir de 0
  For I = 0 To Icc Step dI
   If p < I * VdeI(I) Then p = VdeI(I) * I
   ' si max actuel< puissance actuelle alors...
  Next
  ' ici p=max de puissance à cette valeur de Rs
 Loop Until p > Pm 'on diminue Rs jusq'à ce que p soit > Pm
 dRs = -dRs / 10 ' changer de sens de déplacement
            ' avec une précision 10 fois meilleure.
Loop While dRs > Rs / 10000
' MsgBox " Rs=" & Rs
truc: Rs = 1
End Sub
Sub TracerCaracteristiqueTheorique()
  Call CalculConstPan
  Y(0) = 0
  X(0) = Voc
  For I = 1 To Ndim - 1
   Y(I) = Icc * I / Ndim ' x c'est la tension. y, le courant.
   X(I) = VdeI(Y(I))
  Next
  X(Ndim) = 0
  Y(Ndim) = Icc
  Call courbe(Module2.X(), Module2.Y(), True, vbBlue)
  osc.Show
End Sub
Sub CalculConstPan()
'  Call calculensol
  TPanneau = TExterne + 30 * EnSol

  Voco = LstVPan(PanActuel)
  Voc = Voco * (1 - Gamma * (TPanneau - T0)) * Log(Exp(1) + Beta * (EnSol - 1#)) '
L'ensoleillement de référence est de 1 KW/m²
  Icco = LstI(PanActuel)
  Icc = Icco * (1 + Alpha * (TPanneau - T0))
  Pm = Voco = LstPPAN(PanActuel) * (Icc * Voc) / (Icco * Voco)
 ' MsgBox "voc=" & Voc & " icc=" & Icc & " pm=" & Pm & " pm/v/i" & Pm / Voc / Icc
  Lamda = 20.7 / Voc
  'MsgBox " I à Pm=" & Im
  'Rs = Pm / Im ^ 2 - Voc / 20.7 / (Icc - Im)
   ChercherRs

End Sub
```

3. *Listing de «module2 » (fonction de traçage de la courbe) :*

```
Public Const Ndim = 1500
Public X(Ndim) As Single, Y(Ndim) As Single, xlog As Single, ylog As Single, xmin As
Single, xmax As Single, ymin As Single, ymax As Single, first As Boolean
Private I As Integer, xx As Single, yy As Single
Private Sub axex()
 ylog = 10 ^ Int(Log(ymax - ymin) / Log(10!))
 ymin = (Int(ymin / ylog)) * ylog
 ymax = (Int(ymax / ylog) + 1) * ylog
 osc.Label2 = Format(ymin, "0.00e+00")
 osc.Label1 = Format(ymax, "0.00e+00")

  For yy = ymin To ymax Step ylog
   ynext = osc.Picture1.Height * (ymax - yy) / (ymax - ymin)
' ici remlacer osc par une autre variable
   If Abs(yy) < ylog / 2! Then
    osc.Picture1.Line (0, ynext)-Step(osc.Picture1.Width, 0), &H80FFFF
   Else
    osc.Picture1.Line (0, ynext)-Step(osc.Picture1.Width, 0), 0
   End If
  Next
End Sub

Private Sub axey()
 xlog = 10 ^ Int(Log(xmax - xmin) / Log(10!))
 xmin = (Int(xmin / xlog)) * xlog
 xmax = (Int(xmax / xlog) + 1) * xlog
 osc.Label3 = Format(xmin, "0.00e+00")
 osc.Label4 = Format(xmax, "0.00e+00")

  For xx = xmin To xmax Step xlog
   xnext = osc.Picture1.Width * (xx - xmin) / (xmax - xmin)
' ici remlacer osc par une autre variable
  If Abs(xx) < xlog / 2! Then
   osc.Picture1.Line (xnext, 0!)-Step(0, osc.Picture1.Height), &H80FFFF
  Else
   osc.Picture1.Line (xnext, 0!)-Step(0, osc.Picture1.Height), 0
  End If
  Next
 End Sub
Sub courbe(X() As Single, Y() As Single, effacer As Boolean, col As Long)

' Calcul de min,max
xmin = X(0)
xmax = X(0)
ymin = Y(0)
ymax = Y(0)
For I = 0 To Ndim
 If X(I) < xmin Then
```

```
 xmin = X(I)
ElseIf X(I) > xmax Then
  xmax = X(I)
End If
If Y(I) < ymin Then
 ymin = Y(I)
ElseIf Y(I) > ymax Then
  ymax = Y(I)
End If
Next I
If xmax <= xmin Then xmax = xmin + 1!
If ymax <= ymin Then ymax = ymin + 1!
' trace de courbe
' ici remlacer osc par une autre variable
'osc.Print first
'If osc.Check1 Then
 axex
 axey
' first = False
'Else
' If Not first Then first = Not first: axex: axey
'End If
osc.Picture1.PSet (osc.Picture1.Width * (X(0) - xmin) / (xmax - xmin), osc.Picture1.Height *
(ymax - Y(0)) / (ymax - ymin))
For I = 0 To Ndim
  xnext = osc.Picture1.Width * (X(I) - xmin) / (xmax - xmin)
  ynext = osc.Picture1.Height * (ymax - Y(I)) / (ymax - ymin)
' ici remlacer osc par une autre variable
  osc.Picture1.Line Step(0, 0)-(xnext, ynext), col
Next I
End Sub
```

4. Listing de «frmSimul» (fenêtre principale) :

```
Option Explicit

Private Sub ActiverBat_Click()
  frmBat.Show
End Sub

Private Sub ActiverConv_Click()
 FrmConv.Show
End Sub

Private Sub ActiverPanneau_Click()
  FrmPan.Show
End Sub

Private Sub Activerpompe_Click()
 FrmPompe.Show
End Sub

Private Sub Activerpui_Click()
 FrmPuiRes.Show
End Sub

Private Sub ActiverRes_Click()
 FrmPuiRes.Show
End Sub

Private Sub Form_Load()
 Call InitGlobal
End Sub

Private Sub Form_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As
Single)
If (X < FrmSimul.Width * 0.3) And (Y > FrmSimul.Height * 0.6) Then
  FrmPan.Show
ElseIf (X < FrmSimul.Width * 0.96) And (X > FrmSimul.Width * 0.77) And (Y <
FrmSimul.Height * 1.21) And (Y > FrmSimul.Height * 0.9) Then
  FrmPompe.Show
ElseIf (X < FrmSimul.Width * 1.16) And (X > FrmSimul.Width * 1.08) And (Y <
FrmSimul.Height * 0.77) And (Y > FrmSimul.Height * 0.55) Then
  FrmPuiRes.Show
ElseIf (X < FrmSimul.Width * 1.18) And (X > FrmSimul.Width * 1.01) And (Y <
FrmSimul.Height * 1.23) And (Y > FrmSimul.Height * 1.01) Then
  FrmPuiRes.Show
End If
'Label1 = x / FrmSimul.Width & " " & y / FrmSimul.Height
End Sub
```

```
Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As
Single)
If (X < Shape1.Left + Shape1.Width) And (X > Shape1.Left) And (Y > Shape1.Top) And (Y
< Shape1.Top + Shape1.Height) And (Button = 1) Then
  Shape3.Top = Y
  Shape3.Height = Shape1.Top + Shape1.Height - Y
End If
  Call changeTExterne((Shape3.Height / Shape1.Height * 120!) - 20!)
  'If Not Frmsimul.Enabled Then Frmsimul.Icon =
LoadPicture("c:\Bouheb~1\images\pompe_~1.bmp")
End Sub

Private Sub Form_Unload(Cancel As Integer)
 End
End Sub

Private Sub Image3_Click()
  frmBat.Show
End Sub

Private Sub Image4_Click()
   FrmConv.Show
End Sub

Private Sub menuabout_Click()
 FrmAbout.Show 1
End Sub

Private Sub MenuDim_Click()
 FrmDim.Show
End Sub

Private Sub menuexit_Click()
 Call Form_Unload(1)
End Sub

Private Sub optionbat_Click()
 optionbat.Checked = Not optionbat.Checked
 Image3.Visible = optionbat.Checked
 Probateries.Visible = optionbat.Checked
End Sub

Private Sub optionconv_Click()
 optionconv.Checked = Not optionconv.Checked
 Image4.Visible = optionconv.Checked
 propconv.Visible = optionconv.Checked
End Sub

Private Sub Probateries_Click()
 PRbat.Show
```

```
End Sub

Public Sub proPanneau_Click()
 'Call PRpan.Form_Load
 PRpan.Show vbModal
End Sub

Public Sub propconv_Click()
 FrmConv.Show
End Sub

Public Sub proppompe_Click()
' MsgBox " pas encore"
 PRpompe.Show vbModal
End Sub
```

5. *Listing de «frmPan » (fenêtre du panneau) :*

```
Option Explicit
Dim BranchePan As Boolean

Private Sub Command1_Click()
Dim I As Integer, pre As Boolean
pre = True
osc.Show
End Sub

Private Sub Command2_Click()
 Call FrmSimul.proPanneau_Click
End Sub

Private Sub Command3_Click()
 sol.Show
End Sub

Private Sub Command4_Click()
 BranchePan = Not BranchePan
 If BranchePan Then Command4.Picture = Image1(0).Picture Else Command4.Picture =
Image1(1).Picture
' SUPan.Show
End Sub

Private Sub Command5_Click()
SUPan.Show
' Histopan.Show
End Sub

Private Sub Form_Load()
 If Not DAQ Then
  Frame1.Enabled = False
  Label1.Enabled = False
```

```
 Label2.Enabled = False
 Label3.Enabled = False
 Label4.Enabled = False
 Label5.Enabled = False
 Label6.Enabled = False
Else
 Frame1.Enabled = True
 Label1.Enabled = True
 Label2.Enabled = True
 Label3.Enabled = True
 Label4.Enabled = True
 Label5.Enabled = True
 Label6.Enabled = True
 Text1 = VPanneau
 Text2 = IPanneau
 Text3 = Format(TPanneau, "###.## °c")
 Text4 = AngleSite
 Text5 = AngleAzimut
End If
Text6 = TypesPAN(PanActuel)
Text7 = Constructeur(PanActuel)
Text8 = LstVPan(PanActuel)
Text9 = LstI(PanActuel)
Text10 = LstPPAN(PanActuel)
Text11 = LstL(PanActuel) * LstW(PanActuel)
Text12 = LstPdPAN(PanActuel)
Text13 = LstPrixPAN(PanActuel)

' Fin = False
End Sub

Private Sub Form_Unload(Cancel As Integer)
' Fin = True
osc.Visible = False
sol.Visible = False
SUPan.Visible = False
End Sub

Private Sub Text2_LostFocus()
'Call changeipanneau(Val(Text2))
End Sub

Private Sub Text3_LostFocus()
' Call changeTPanneau(Val(Text3))
End Sub
```

6. Listing de «PRPan » (fenêtre des propriétés du paneau) :

```
Option Explicit
Private s As Integer, List1HasFocus As Boolean
Function contient(l As ListBox, s As String) As Boolean
```

```
 Dim I As Integer
 For I = 0 To l.ListCount - 1
 If UCase(l.List(I)) = UCase(s) Then Exit For
 Next
 contient = (I < l.ListCount)
End Function
Private Sub CmAnnuler_Click()
 Me.Hide
End Sub

Private Sub CmOk_Click()
 Me.Hide
End Sub

Private Sub CmNouveau_Click()
 Dim d As String, I As Integer
 If List1HasFocus Then
   d = InputBox("Nom du constructeur ?", "Nouveau constructeur")
   If (d <> "") And (Not contient(List1, d)) Then
    List1.AddItem d
    ndxLstPan = ndxLstPan + 1
    ReDim Preserve IdxLstPan(ndxLstPan)
    IdxLstPan(ndxLstPan) = NLstPan
    'ReDim Preserve TypesPAN(NLstPan)
    Call List1_Click
   End If
 Else
  d = InputBox("Type de panneau ?", "Nouveau panneau")
  If (d <> "") And (Not contient(List2, d)) Then
   ReDim Preserve TypesPAN(NLstPan)
   ReDim Preserve LstVPan(NLstPan)
   ReDim Preserve LstI(NLstPan)
   ReDim Preserve LstPPAN(NLstPan)
   ReDim Preserve LstW(NLstPan)
   ReDim Preserve LstL(NLstPan)
   ReDim Preserve LstPdPAN(NLstPan)
   ReDim Preserve LstPrixPAN(NLstPan)

   For I = NLstPan To IdxLstPan(List1.ListIndex + 1) Step -1
    TypesPAN(I) = TypesPAN(I - 1)
    LstVPan(I) = LstVPan(I - 1)
    LstI(I) = LstI(I - 1)
    LstPPAN(I) = LstPPAN(I - 1)
    LstL(I) = LstL(I - 1)
    LstW(I) = LstW(I - 1)
    LstPdPAN(I) = LstPdPAN(I - 1)
    LstPrixPAN(I) = LstPrixPAN(I - 1)
   Next
   TypesPAN(IdxLstPan(List1.ListIndex + 1)) = d
   LstVPan(IdxLstPan(List1.ListIndex + 1)) = Val(InputBox("Vco =?"))
```

```
    LstI(IdxLstPan(List1.ListIndex + 1)) = Val(InputBox("Icc =?"))
    LstPPAN(IdxLstPan(List1.ListIndex + 1)) = Val(InputBox("puissance =?"))
    LstL(IdxLstPan(List1.ListIndex + 1)) = Val(InputBox("longeur =?"))
    LstW(IdxLstPan(List1.ListIndex + 1)) = Val(InputBox("largeur =?"))
    LstPdPAN(IdxLstPan(List1.ListIndex + 1)) = Val(InputBox("poids =?"))
    LstPrixPAN(IdxLstPan(List1.ListIndex + 1)) = Val(InputBox("prixs =?"))

    For I = List1.ListIndex + 1 To ndxLstPan
     IdxLstPan(I) = IdxLstPan(I) + 1
    Next
    NLstPan = NLstPan + 1
    Call List1_Click
   End If
  End If
End Sub

Private Sub CmSuppr_Click()
 Dim I As Integer, Idx As Integer, Incr As Integer
 If List1HasFocus Then
   Idx = List1.ListIndex
   If List1.ListCount = 1 Then MsgBox (List1 & " ne peut etre supprimé !"): Exit Sub
   Incr = IdxLstPan(Idx + 1) - IdxLstPan(Idx)
   If MsgBox(List1 & " sera supprimé avec tous ses paneaux." & Chr(13) & " Supprimer
comme même ?", vbYesNo + vbExclamation, "Supprimer un constructeur") = vbYes Then
    For I = IdxLstPan(Idx) To NLstPan - Incr - 1
     TypesPAN(I) = TypesPAN(I + Incr)
     LstVPan(I) = LstVPan(I + Incr)
     LstI(I) = LstI(I + Incr)
     LstPPAN(I) = LstPPAN(I + Incr)
     LstL(I) = LstL(I + Incr)
     LstW(I) = LstW(I + Incr)
     LstPdPAN(I) = LstPdPAN(I + Incr)
     LstPrixPAN(I) = LstPrixPAN(I + Incr)
    Next
    NLstPan = NLstPan - Incr
    ReDim Preserve TypesPAN(NLstPan - 1)
    ReDim Preserve LstVPan(NLstPan - 1)
    ReDim Preserve LstI(NLstPan - 1)
    ReDim Preserve LstPPAN(NLstPan - 1)
    ReDim Preserve LstL(NLstPan - 1)
    ReDim Preserve LstW(NLstPan - 1)
    ReDim Preserve LstPdPAN(NLstPan - 1)
    ReDim Preserve LstPrixPAN(NLstPan - 1)

    ndxLstPan = ndxLstPan - 1
    For I = Idx To ndxLstPan
     IdxLstPan(I) = IdxLstPan(I + 1) - Incr
    Next
    ReDim Preserve IdxLstPan(ndxLstPan)
    List1.RemoveItem (Idx)
```

```
   Call List1_Click
  End If
 Else
  Idx = List1.ListIndex
  If MsgBox("supprimer " & List2 & "  ?", vbYesNo + vbExclamation, "Supprimer un
panneau") = vbYes Then
   For I = Idx + 1 To ndxLstPan
    IdxLstPan(I) = IdxLstPan(I) - 1
   Next
   Idx = IdxLstPan(Idx) + List2.ListIndex
   NLstPan = NLstPan - 1
   For I = Idx To NLstPan - 1
    TypesPAN(I) = TypesPAN(I + 1)
    LstVPan(I) = LstVPan(I + 1)
    LstI(I) = LstI(I + 1)
    LstPPAN(I) = LstPPAN(I + 1)
    LstL(I) = LstL(I + 1)
    LstW(I) = LstW(I + 1)
    LstPdPAN(I) = LstPdPAN(I + 1)
    LstPrixPAN(I) = LstPrixPAN(I + 1)
   Next
   Call List1_Click
  End If
 End If
End Sub

Public Sub Form_Load()
 For s = 0 To 2
  Frame1(s).Top = TabStrip1.ClientTop
  Frame1(s).Left = TabStrip1.ClientLeft
 Next
 Option2 = True 'non orientable
 'on masque les autres options
 Option1 = False
 CheckAzimut.Enabled = False
 CheckSite.Enabled = False
 Frame6.Enabled = False
 Frame1(0).ZOrder (0)
End Sub

Public Sub Form_Unload(Cancel As Integer)
'f
End Sub

Private Sub List1_Click()
 Dim I  As Integer
 If List1.ListIndex < 0 Then List1.ListIndex = 0
 List2.Clear
 I = IdxLstPan(List1.ListIndex)
 Do While (IdxLstPan(List1.ListIndex + 1) > I)
```

```
 List2.AddItem TypesPAN(I)
 I = I + 1
 Loop
 List1HasFocus = True
End Sub

Private Sub List2_Click()
 If List2.ListIndex > -1 Then
  TxtV = LstVPan(IdxLstPan(List1.ListIndex) + List2.ListIndex)
  TxtI = LstI(IdxLstPan(List1.ListIndex) + List2.ListIndex)
  TxtP = LstPPAN(IdxLstPan(List1.ListIndex) + List2.ListIndex)
  TxtL = LstL(IdxLstPan(List1.ListIndex) + List2.ListIndex)
  TxtW = LstW(IdxLstPan(List1.ListIndex) + List2.ListIndex)
  TxtS = TxtL * TxtW
  TxtPoids = LstPdPAN(IdxLstPan(List1.ListIndex) + List2.ListIndex)
  TxtPrix = LstPrixPAN(IdxLstPan(List1.ListIndex) + List2.ListIndex)
  TxtRend = LstRendPan(IdxLstPan(List1.ListIndex) + List2.ListIndex)
 End If
End Sub

Private Sub List2_GotFocus()
 List1HasFocus = False
 If List2.ListIndex > -1 Then
  CmSuppr.Enabled = True
 Else
  CmSuppr.Enabled = False
 End If
End Sub

Private Sub List1_GotFocus()
 List1HasFocus = True
 If List1.ListIndex > -1 Then
  CmSuppr.Enabled = True
 Else
  CmSuppr.Enabled = False
 End If
End Sub

Private Sub Option1_Click()
 Option2 = False
 Option1 = True
 Frame6.Enabled = True
 CheckAzimut.Enabled = True
 CheckSite.Enabled = True
End Sub

Private Sub Option2_Click()
 Option1 = False
 Option2 = True
 Frame6.Enabled = False
```

```
 CheckAzimut.Enabled = False
 CheckSite.Enabled = False
End Sub

Private Sub TabStrip1_Click()
 Frame1(TabStrip1.SelectedItem.Index - 1).ZOrder 0
 If TabStrip1.SelectedItem.Index = 2 And List2.ListIndex < 0 Then List2.Selected(0) = True
End Sub
```

7. Listing de «Sol » (fenêtre de simulation du soleil et l'ensoleillement) :

```
Option Explicit
Private ne As Boolean, Dte As Date
Private I(3) As Double
' un tableau pour enregistrer le numéro du premier jour de chaque mois.


Sub CalculPositionSoleil(Dte As Date, I() As Double)
 Dim b As Double, t As Double, A As Double, l As Double
 Dim a1(3, 3) As Double, a2(3, 3) As Double, a3(3, 3) As Double
 Dim i1(3) As Double, i2(3) As Double, i3(3) As Double
 ne = False
 b = (23 + 27 / 60) * Pi / 180
 t = (Day(Dte) + NJours(Month(Dte)) - 80) * 2 * Pi / 365
 A = 2 * Pi * ((Hour(Dte) - 12 + Minute(Dte) / 60# + Second(Dte) / 3600#) / 24 - (Day(Dte) +
NJours(Month(Dte)) - 80) / 365)
 l = 37 / 180 * Pi ' position d'Alger

 i1(1) = Cos(t)
 i1(2) = Sin(t)
 i1(3) = 0

 a1(1, 1) = 1
 a1(2, 2) = Cos(b)
 a1(3, 3) = Cos(b)
 a1(2, 3) = -Sin(b)
 a1(3, 2) = Sin(b)

 a2(3, 3) = 1
 a2(1, 1) = Cos(A)
 a2(2, 2) = Cos(A)
 a2(1, 2) = -Sin(A)
 a2(2, 1) = Sin(A)

 a3(2, 2) = 1
 a3(1, 3) = -Cos(l)
 a3(3, 1) = Cos(l)
 a3(1, 1) = Sin(l)
 a3(3, 3) = Sin(l)

 Call mul(a1(), i1(), i2())
```

```
'Print "x="; Format(i2(1), "0.00000"); " y="; Format(i2(2), "0.00000"); " z="; Format(i2(3),
"0.00000"), "Le "; Text1; " eme jour à"; Text2; " heures"
Call mul(a2(), i2(), i3())
'Print "x="; Format(i3(1), "0.00000"); " y="; Format(i3(2), "0.00000"); " z="; Format(i3(3),
"0.00000"), "Le "; Text1; " eme jour à"; Text2; " heures"
Call mul(a3(), i3(), I())
'Print "x="; Format(i(1), "0.00000"); " y="; Format(i(2), "0.00000"); " z="; Format(i(3),
"0.00000"), "Le "; Text1; " eme jour à"; Text2; " heures"
'Print i(1) ^ 2 + i(2) ^ 2 + i(3) ^ 2
l = 2000 'Sqr((Line4.X2 - Line4.X1) ^ 2 + (Line4.Y2 - Line4.Y1) ^ 2)
Line4.X2 = Line4.X1 + (-l * I(1) + l * I(2)) * Cos(Pi / 6)
Line4.Y2 = Line4.Y1 + (l * I(1) + l * I(2)) * Sin(Pi / 6) - l * I(3)
Line5.X2 = Line4.X2
Line5.X1 = Line4.X1
Line5.Y1 = Line4.Y1
Line5.Y2 = Line4.Y1 + (l * I(1) + l * I(2)) * Sin(Pi / 6)

Line6.X2 = Line4.X2
Line6.X1 = Line4.X2
Line6.Y1 = Line4.Y2
Line6.Y2 = Line6.Y1 + l * I(3)
Shape1.Left = Line4.X2 - Shape1.Width / 2
Shape1.Top = Line4.Y2 - Shape1.Height / 2
If I(3) < 0 Then Shape1.Visible = False Else Shape1.Visible = True
End Sub
Sub mul(m() As Double, v1() As Double, v2() As Double)
Dim I As Integer, J As Integer

For I = 1 To 3
 v2(I) = 0
 For J = 1 To 3
  v2(I) = m(I, J) * v1(J) + v2(I)
 Next
Next

End Sub

Private Sub affiche()
 Dim r As Double
 Call CalculPositionSoleil(Now, I())
 r = Sqr(I(1) ^ 2 + I(2) ^ 2)
 If r > 0 Then Text3 = CInt(Atn(I(3) / r) * 180 / Pi) & "°" Else Text3 = 90 & "°"

 r = CInt(Atn(I(2) / I(1)) * 180 / Pi)
 If I(1) < 0 Then If I(2) < 0 Then r = r - 180 Else r = r + 180
 Text4 = r & "°"
End Sub

Private Sub Form_Load()
 'calcul du numero de jour du premier de chaque mois.
```

```
 Dte = Now
 txtjour = Day(Dte): Jj.Value = txtjour
 txtmois = Month(Dte): Mm.Value = txtmois
 txtan = Year(Dte): An.Value = txtan
 txtheure = Hour(Dte): Hr.Value = txtheure
 Txtminute = Minute(Dte): Mn.Value = Txtminute
 txtseconde = Second(Dte): Sec.Value = txtseconde
 affiche
 Dim l As Single
 l = Sqr((Line2.X2 - Line2.X1) ^ 2 + (Line2.Y2 - Line2.Y1) ^ 2)
 Line2.X2 = Line2.X1 + l * Cos(Pi / 6)
 Line2.Y2 = Line2.Y1 + l * Sin(Pi / 6)
 Line1.X2 = Line1.X1 - l * Cos(Pi / 6)
 Line1.Y2 = Line1.Y1 + l * Sin(Pi / 6)
End Sub

Public Sub Form_Unload(Cancel As Integer)
' rien
End Sub

Private Sub Txtheure_Change()
'Dim s As String
' s = Text1
' On Error GoTo erd
If txtheure = "" Then txtheure = "0"
If sol.Visible = True Then Dte = CDate(txtjour & "/" & txtmois & "/" & txtan) +
CDate(txtheure & ":" & Txtminute & ":" & txtseconde)
 'Text5 = Format(Dte, "dd/mm/yyyy hh:mm:ss")
'erd:
' If Err = 13 Then Text1 = s: Resume
affiche
End Sub

Private Sub Txtminute_Change()
 'Dim s As String
 's = Text2
' On Error GoTo erd
If Txtminute = "" Then Txtminute = "0"
If sol.Visible = True Then Dte = CDate(txtjour & "/" & txtmois & "/" & txtan) +
CDate(txtheure & ":" & Txtminute & ":" & txtseconde)
 'Text5 = Format(Dte, "dd/mm/yyyy hh:mm:ss")
'erd:
' If Err = 13 Then Text2 = s: Resume
affiche
End Sub


Private Sub Txtseconde_Change()
 'Dim s As String
```

```
's = Text2
' On Error GoTo erd
 If txtseconde = "" Then txtseconde = "0"
 Dte = CDate(txtjour & "/" & txtmois & "/" & txtan) + CDate(txtheure & ":" & Txtminute &
":" & txtseconde)
 'Text5 = Format(Dte, "dd/mm/yyyy hh:mm:ss")
'erd:
'  If Err = 13 Then Text2 = s: Resume
affiche
End Sub

Private Sub Txtan_Change()
' Dim s As String
' s = Text2
' On Error GoTo erd
If txtan = "" Then txtan = "0"
 Dte = CDate(txtjour & "/" & txtmois & "/" & txtan) + CDate(txtheure & ":" & Txtminute &
":" & txtseconde)
 'Text5 = Format(Dte, "dd/mm/yyyy hh:mm:ss")
'erd:
'  If Err = 13 Then Text2 = s: Resume
affiche
End Sub

Private Sub Txtmois_Change()
' Dim s As String
 's = Text2
 'On Error GoTo erd
 If txtmois = "" Then txtmois = "0"
 If txtmois = "2" And txtjour > 28 Then If txtan Mod 4 <> 0 Then txtjour = 28 Else txtjour =
29
 If txtmois = "4" And txtjour > 30 Then txtjour = 30
 If txtmois = "6" And txtjour > 30 Then txtjour = 30
 If txtmois = "11" And txtjour > 30 Then txtjour = 30
 If txtmois = "9" And txtjour > 30 Then txtjour = 30
 If sol.Visible = True Then Dte = CDate(txtjour & "/" & txtmois & "/" & txtan) +
CDate(txtheure & ":" & Txtminute & ":" & txtseconde)
 'Text5 = Format(Dte, "dd/mm/yyyy hh:mm:ss")
'erd:
'  If Err = 13 Then Text2 = s: Resume
affiche
End Sub


Private Sub Txtjour_Change()
 'Dim s As String
 's = Text2
 'On Error GoTo erd
 If txtjour = "" Then txtjour = "0"
```

```
 If txtmois = "2" And txtjour > 28 Then If txtan Mod 4 <> 0 Then txtjour = 28 Else txtjour =
29
 If txtmois = "4" And txtjour > 30 Then txtjour = 30
 If txtmois = "6" And txtjour > 30 Then txtjour = 30
 If txtmois = "11" And txtjour > 30 Then txtjour = 30
 If txtmois = "9" And txtjour > 30 Then txtjour = 30
 If sol.Visible = True Then Dte = CDate(txtjour & "/" & txtmois & "/" & txtan) +
CDate(txtheure & ":" & Txtminute & ":" & txtseconde)
 'Text5 = Format(Dte, "dd/mm/yyyy hh:mm:ss")
'erd:
'  If Err = 13 Then Text2 = s: Resume
affiche
End Sub
```

8. *Listing d' «osc » (fenêtre de l'oscilloscope – courbe I-V du panneau) :*

```
Option Explicit
Private n As Integer
Private Sub Form_Load()
n = 1
Text1 = TExterne
Text2 = EnSol
Me.Show
Call TracerCaracteristiqueTheorique
End Sub

Public Sub Form_Unload(Cancel As Integer)
'rien
End Sub

Private Sub Picture1_LostFocus()
 osc.Line1.BorderStyle = 0
 osc.Line2.BorderStyle = 0
End Sub
Private Sub Picture1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As
Single)
' déplacer les curseurs si necessaire.
 Dim xx As Single, I As Integer, yy As Single
If Button = 0 Then
 osc.Line1.BorderStyle = 0
 osc.Line2.BorderStyle = 0
 osc.Label5 = " x=" & Format((((xmax - xmin) * (X / Picture1.Width) + xmin)), "0.000 ")
 osc.Label6 = " y=" & Format((((ymax - ymin) * ((Picture1.Height - Y) / Picture1.Height) +
ymin)), "0.000 ")
ElseIf X > 0 And X < Picture1.Width And Button = 1 Then
 osc.Line1.X1 = X
 osc.Line1.X2 = X
 osc.Line1.BorderStyle = 3
 xx = xmin + (xmax - xmin) * (X / Picture1.Width)
 For I = n To Ndim
```

```
 If Module2.X(I) < xx And xx < Module2.X(I - 1) Or Module2.X(I - 1) < xx And xx <
Module2.X(I) Then n = I
 Next
 For I = 1 To n - 1
 If Module2.X(I) < xx And xx < Module2.X(I - 1) Or Module2.X(I - 1) < xx And xx <
Module2.X(I) Then n = I
 Next

 osc.Label5 = " x=" & Format(xx, "0.000 ")
 yy = (Module2.Y(n))
 osc.Label6 = " y=" & Format(yy, "0.000")
 osc.Line2.Y1 = osc.Picture1.Height * (ymax - yy) / (ymax - ymin)
 osc.Line2.Y2 = osc.Line2.Y1
 osc.Line2.BorderStyle = 3
End If
End Sub
```

9. Listing de «frmPompe » (fenêtre de la pompe) :

```
Option Explicit
Private Sub Command1_Click()
 Call FrmSimul.proppompe_Click ' appelle le menu qui à son tour appelle la bonne fenetre
End Sub
```

10. Listing de «PRPompe » (fenêtre des propriétés de la pompe) :

```
Option Explicit
Private s As Integer, List1HasFocus As Boolean
Function contient(l As ListBox, s As String) As Boolean
 Dim I As Integer
 For I = 0 To l.ListCount - 1
 If UCase(l.List(I)) = UCase(s) Then Exit For
 Next
 contient = (I < l.ListCount)
End Function
Private Sub CmAnnuler_Click()
 Me.Hide
End Sub

Private Sub CmOk_Click()
 Me.Hide
End Sub

Private Sub CmNouveau_Click()
 Dim d As String, I As Integer
 If List1HasFocus Then
   d = InputBox("Nom du constructeur ?", "Nouveau constructeur")
   If (d <> "") And (Not contient(List1, d)) Then
    List1.AddItem d
    NdxLst = NdxLst + 1
    ReDim Preserve IdxLstPompe(NdxLst)
    IdxLstPompe(NdxLst) = NLst
```

```
    ReDim Preserve Typespompe(NLst)
    Call List1_Click
   End If
  Else
   d = InputBox("Type de pompe ?", "Nouvelle pompe")
   If (d <> "") And (Not contient(List2, d)) Then
   ReDim Preserve TypesPompe(NLstPompe)
   ReDim Preserve LstVPompe(NLstPompe)
   ReDim Preserve LstPPompe(NLstPompe)
   ReDim Preserve LstRendPompe(NLstPompe)
   ReDim Preserve LstDPompe(NLstPompe)
   ReDim Preserve LstPrixPompe(NLstPompe)
   ReDim Preserve LstImmersion(NLstPompe)

'couper coller fin.
    For I = NLstPompe To IdxLstPompe(List1.ListIndex + 1) Step -1
      TypesPompe(I) = TypesPompe(I - 1)
      LstVPompe(I) = LstVPompe(I - 1)
      LstPPompe(I) = LstPPompe(I - 1)
      LstRendPompe(I) = LstRendPompe(I - 1)
      LstDPompe(I) = LstDPompe(I - 1)
      LstPrixPompe(I) = LstPrixPompe(I - 1)
      LstImmersion(I) = LstImmersion(I - 1)
    Next
    TypesPompe(IdxLstPompe(List1.ListIndex + 1)) = d
    LstVPompe(IdxLstPompe(List1.ListIndex + 1)) = Val(InputBox("V nominale =?"))
    LstPPompe(IdxLstPompe(List1.ListIndex + 1)) = Val(InputBox("puissance =?"))
    LstRendPompe(IdxLstPompe(List1.ListIndex + 1)) = Val(InputBox("Rendement =?"))
    LstDPompe(IdxLstPompe(List1.ListIndex + 1)) = Val(InputBox("Diamètre =?"))
    LstPrixPompe(IdxLstPompe(List1.ListIndex + 1)) = Val(InputBox("prix =?"))
    LstImmersion(IdxLstPompe(List1.ListIndex + 1)) = Val(InputBox("Hauteur relativement à
la surface du liquide (Algébrique) =?"))

    For I = List1.ListIndex + 1 To NdxLst
     IdxLstPompe(I) = IdxLstPompe(I) + 1
    Next
    NLst = NLst + 1
    Call List1_Click
   End If
  End If
End Sub

Private Sub CmSuppr_Click()
 Dim I As Integer, Idx As Integer, Incr As Integer
 If List1HasFocus Then
   Idx = List1.ListIndex
   If List1.ListCount = 1 Then MsgBox (List1 & " ne peut etre supprimé !"): Exit Sub
   Incr = IdxLstPompe(Idx + 1) - IdxLstPompe(Idx)
   If MsgBox(List1 & " sera supprimé avec toutes ses pompes." & Chr(13) & " Supprimer
comme même ?", vbYesNo + vbExclamation, "Supprimer un constructeur") = vbYes Then
```

```
    For I = IdxLstPompe(Idx) To NLst - Incr - 1
     TypesPompe(I) = TypesPompe(I + Incr)
     LstVPompe(I) = LstVPompe(I + Incr)
     LstPPompe(I) = LstPPompe(I + Incr)
     LstRendPompe(I) = LstRendPompe(I + Incr)
     LstDPompe(I) = LstDPompe(I + Incr)
     LstPrixPompe(I) = LstPrixPompe(I + Incr)
     LstImmersion(I) = LstImmersion(I + Incr)
    Next
    NLst = NLst - Incr
    ReDim Preserve TypesPompe(NLst - 1)
    ReDim Preserve LstVPompe(NLst - 1)
    ReDim Preserve LstPPompe(NLst - 1)
    ReDim Preserve LstRendPompe(NLst - 1)
    ReDim Preserve LstDPompe(NLst - 1)
    ReDim Preserve LstPrixPompe(NLst - 1)
    ReDim Preserve LstImmersion(NLst - 1)

    NdxLst = NdxLst - 1
    For I = Idx To NdxLst
     IdxLstPompe(I) = IdxLstPompe(I + 1) - Incr
    Next
    ReDim Preserve IdxLstPompe(NdxLst)
    List1.RemoveItem (Idx)
    Call List1_Click
   End If
  Else
   Idx = List1.ListIndex
   If MsgBox("supprimer " & List2 & "  ?", vbYesNo + vbExclamation, "Supprimer une
pompe") = vbYes Then
    For I = Idx + 1 To NdxLst
     IdxLstPompe(I) = IdxLstPompe(I) - 1
    Next
    Idx = IdxLstPompe(Idx) + List2.ListIndex
    NLst = NLst - 1
    For I = Idx To NLst - 1
     TypesPompe(I) = TypesPompe(I + 1)
     LstVPompe(I) = LstVPompe(I + 1)
     LstPPompe(I) = LstPPompe(I + 1)
     LstRendPompe(I) = LstRendPompe(I + 1)
     LstDPompe(I) = LstDPompe(I + 1)
     LstImmersion(I) = LstImmersion(I + 1)
     LstPrixPompe(I) = LstPrixPompe(I + 1)
    Next
    Call List1_Click
   End If
  End If
End Sub

Public Sub Form_Load()
```

```
For s = 0 To 2
 Frame1(s).Top = TabStrip1.ClientTop
 Frame1(s).Left = TabStrip1.ClientLeft
Next
Option2 = True 'non orientable
'on masque les autres options
Option1 = False
Frame1(0).ZOrder (0)
End Sub

Public Sub Form_Unload(Cancel As Integer)
'f
End Sub

Private Sub List1_Click()
 Dim I  As Integer
 If List1.ListIndex < 0 Then List1.ListIndex = 0
 List2.Clear
 I = IdxLstPompe(List1.ListIndex)
 Do While (IdxLstPompe(List1.ListIndex + 1) > I)
  List2.AddItem TypesPompe(I)
  I = I + 1
 Loop
 List1HasFocus − True
End Sub

Private Sub List2_Click()
 If List2.ListIndex > -1 Then
  TxtV − LstVPompe(IdxLstPompe(List1.ListIndex) + List2.ListIndex)
  TxtP = LstPPompe(IdxLstPompe(List1.ListIndex) + List2.ListIndex)
  TxtDiametre = LstDPompe(IdxLstPompe(List1.ListIndex) + List2.ListIndex)
  TxtPriX = LstPrixPompe(IdxLstPompe(List1.ListIndex) + List2.ListIndex)
  TxtRendement = LstRendPompe(IdxLstPompe(List1.ListIndex) + List2.ListIndex)
  TxtProfo = Abs(LstImmersion(IdxLstPompe(List1.ListIndex) + List2.ListIndex))
  If LstImmersion(IdxLstPompe(List1.ListIndex) + List2.ListIndex) > 0 Then Call
Option2_Click Else Call Option1_Click
 End If
End Sub

Private Sub List2_GotFocus()
 List1HasFocus = False
 If List2.ListIndex > -1 Then
  CmSuppr.Enabled = True
 Else
  CmSuppr.Enabled − False
 End If
End Sub

Private Sub List1_GotFocus()
 List1HasFocus = True
```

```
If List1.ListIndex > -1 Then
 CmSuppr.Enabled = True
Else
 CmSuppr.Enabled = False
End If
End Sub

Private Sub Option1_Click()
 Option2 = False
 Option1 = True
End Sub

Private Sub Option2_Click()
 Option1 = False
 Option2 = True
End Sub

Private Sub TabStrip1_Click()
 Frame1(TabStrip1.SelectedItem.Index - 1).ZOrder 0
 If TabStrip1.SelectedItem.Index = 2 And List2.ListIndex < 0 Then List2.Selected(0) = True
End Sub
```

11. Listing de «PRBat » (fenêtre des propriétés de la batterie) :

```
Option Explicit
Private s As Integer, List1HasFocus As Boolean
Function contient(l As ListBox, s As String) As Boolean
 Dim I As Integer
 For I = 0 To l.ListCount - 1
 If UCase(l.List(I)) = UCase(s) Then Exit For
 Next
 contient = (I < l.ListCount)
End Function
Private Sub CmAnnuler_Click()
 Me.Hide
End Sub

Private Sub CmOk_Click()
 Me.Hide
End Sub

Private Sub CmNouveau_Click()
 Dim d As String, I As Integer
 If List1HasFocus Then
   d = InputBox("Nom du constructeur ?", "Nouveau constructeur")
   If (d <> "") And (Not contient(List1, d)) Then
    List1.AddItem d
    ndxLstBat = ndxLstBat + 1
    ReDim Preserve IdxLstBat(ndxLstBat)
    IdxLstBat(ndxLstBat) = NLstBat
```

```
   'ReDim Preserve Typesbat(NLstbat)
    Call List1_Click
   End If
  Else
   d = InputBox("Type de batneau ?", "Nouveau batneau")
   If (d <> "") And (Not contient(List2, d)) Then
    ReDim Preserve TypesBat(NLstBat)
    ReDim Preserve LstVnBat(NLstBat)
    ReDim Preserve LstVminBat(NLstBat)
    ReDim Preserve LstVmaxBat(NLstBat)
    ReDim Preserve LstCmaxBat(NLstBat)
    ReDim Preserve LstCminBat(NLstBat)
    ReDim Preserve LstRendBat(NLstBat)
    ReDim Preserve LstPrixBat(NLstBat)

    For I = NLstBat To IdxLstBat(List1.ListIndex + 1) Step -1
     TypesBat(I) = TypesBat(I - 1)
     LstVnBat(I) = LstVnBat(I - 1)
     LstVminBat(I) = LstVminBat(I - 1)
     LstVmaxBat(I) = LstVmaxBat(I - 1)
     LstCminBat(I) = LstCminBat(I - 1)
     LstCmaxBat(I) = LstCmaxBat(I - 1)
     LstRendBat(I) = LstRendBat(I - 1)
     LstPrixBat(I) = LstPrixBat(I - 1)
    Next
    TypesBat(IdxLstBat(List1.ListIndex + 1)) = d
    LstVnBat(IdxLstBat(List1.ListIndex + 1)) = Val(InputBox("Vco =?"))
    LstVminBat(IdxLstBat(List1.ListIndex + 1)) = Val(InputBox("Icc =?"))
    LstVmaxBat(IdxLstBat(List1.ListIndex + 1)) = Val(InputBox("puissance =?"))
    LstCminBat(IdxLstBat(List1.ListIndex + 1)) = Val(InputBox("longeur =?"))
    LstCmaxBat(IdxLstBat(List1.ListIndex + 1)) = Val(InputBox("largeur =?"))
    LstRendBat(IdxLstBat(List1.ListIndex + 1)) = Val(InputBox("poids =?"))
    LstPrixBat(IdxLstBat(List1.ListIndex + 1)) = Val(InputBox("prixs =?"))

    For I = List1.ListIndex + 1 To ndxLstBat
     IdxLstBat(I) = IdxLstBat(I) + 1
    Next
    NLstBat = NLstBat + 1
    Call List1_Click
   End If
  End If
End Sub

Private Sub CmSuppr_Click()
 Dim I As Integer, Idx As Integer, Incr As Integer
 If List1HasFocus Then
   Idx = List1.ListIndex
   If List1.ListCount = 1 Then MsgBox (List1 & " ne peut etre supprimé !"): Exit Sub
   Incr = IdxLstBat(Idx + 1) - IdxLstBat(Idx)
```

```
  If MsgBox(List1 & " sera supprimé avec tous ses bateaux." & Chr(13) & " Supprimer
comme même ?", vbYesNo + vbExclamation, "Supprimer un constructeur") = vbYes Then
   For I = IdxLstBat(Idx) To NLstBat - Incr - 1
    TypesBat(I) = TypesBat(I + Incr)
    LstVnBat(I) = LstVnBat(I + Incr)
    LstVminBat(I) = LstVminBat(I + Incr)
    LstVmaxBat(I) = LstVmaxBat(I + Incr)
    LstCminBat(I) = LstCminBat(I + Incr)
    LstCmaxBat(I) = LstCmaxBat(I + Incr)
    LstRendBat(I) = LstRendBat(I + Incr)
    LstPrixBat(I) = LstPrixBat(I + Incr)
   Next
   NLstBat = NLstBat - Incr
   ReDim Preserve TypesBat(NLstBat - 1)
   ReDim Preserve LstVnBat(NLstBat - 1)
   ReDim Preserve LstVminBat(NLstBat - 1)
   ReDim Preserve LstVmaxBat(NLstBat - 1)
   ReDim Preserve LstCminBat(NLstBat - 1)
   ReDim Preserve LstCmaxBat(NLstBat - 1)
   ReDim Preserve LstRendBat(NLstBat - 1)
   ReDim Preserve LstPrixBat(NLstBat - 1)

   ndxLstBat = ndxLstBat - 1
   For I = Idx To ndxLstBat
    IdxLstBat(I) = IdxLstBat(I + 1) - Incr
   Next
   ReDim Preserve IdxLstBat(ndxLstBat)
   List1.RemoveItem (Idx)
   Call List1_Click
  End If
 Else
  Idx = List1.ListIndex
  If MsgBox("supprimer " & List2 & "  ?", vbYesNo + vbExclamation, "Supprimer un
batneau") = vbYes Then
   For I = Idx + 1 To ndxLstBat
    IdxLstBat(I) = IdxLstBat(I) - 1
   Next
   Idx = IdxLstBat(Idx) + List2.ListIndex
   NLstBat = NLstBat - 1
   For I = Idx To NLstBat - 1
    TypesBat(I) = TypesBat(I + 1)
    LstVnBat(I) = LstVnBat(I + 1)
    LstVminBat(I) = LstVminBat(I + 1)
    LstVmaxBat(I) = LstVmaxBat(I + 1)
    LstCminBat(I) = LstCminBat(I + 1)
    LstCmaxBat(I) = LstCmaxBat(I + 1)
    LstRendBat(I) = LstRendBat(I + 1)
    LstPrixBat(I) = LstPrixBat(I + 1)
   Next
   Call List1_Click
```

```
  End If
 End If
End Sub

Public Sub Form_Load()
 For s = 0 To 1
  Frame1(s).Top = TabStrip1.ClientTop
  Frame1(s).Left = TabStrip1.ClientLeft
 Next
 'on masque les autres options
 Frame1(0).ZOrder (0)
End Sub

Public Sub Form_Unload(Cancel As Integer)
'f
End Sub

Private Sub List1_Click()
 Dim I  As Integer
 If List1.ListIndex < 0 Then List1.ListIndex = 0
 List2.Clear
 I = IdxLstBat(List1.ListIndex)
 Do While (IdxLstBat(List1.ListIndex + 1) > I)
  List2.AddItem TypesBat(I)
  I = I + 1
 Loop
 List1HasFocus = True
End Sub

Private Sub List2_Click()
 If List2.ListIndex > -1 Then
  TxtVn = LstVnBat(IdxLstBat(List1.ListIndex) + List2.ListIndex)
  TxtVmin = LstVminBat(IdxLstBat(List1.ListIndex) + List2.ListIndex)
  TxtVmax = LstVmaxBat(IdxLstBat(List1.ListIndex) + List2.ListIndex)
  TxtCmin = LstCminBat(IdxLstBat(List1.ListIndex) + List2.ListIndex)
  TxtCmax = LstCmaxBat(IdxLstBat(List1.ListIndex) + List2.ListIndex)
  TxtRendement = LstRendBat(IdxLstBat(List1.ListIndex) + List2.ListIndex)
  TxtPriX = LstPrixBat(IdxLstBat(List1.ListIndex) + List2.ListIndex)
 End If
End Sub

Private Sub List2_GotFocus()
 List1HasFocus = False
 If List2.ListIndex > -1 Then
  CmSuppr.Enabled = True
 Else
  CmSuppr.Enabled = False
 End If
End Sub
```

```
Private Sub List1_GotFocus()
 List1HasFocus = True
 If List1.ListIndex > -1 Then
  CmSuppr.Enabled = True
 Else
  CmSuppr.Enabled = False
 End If
End Sub


Private Sub TabStrip1_Click()
 Frame1(TabStrip1.SelectedItem.Index - 1).ZOrder 0
 If TabStrip1.SelectedItem.Index = 2 And List2.ListIndex < 0 Then List2.Selected(0) = True
End Sub
```

12. Listing de «frmPuiRes » (fenêtre du puits avec le réservoir) :

```
Dim Act As Single
Private Sub Command1_Click()
 Me.Visible = False
End Sub

Private Sub Slider1_Change()
 TxtProf = Slider1
If CSng(TxtNiveau) >= CSng(TxtProf) Then
  Message = "Alerte !" + Chr(13) + "Puits vide"
 ElseIf CSng(TxtNiveau) < CSng(Txtcritique) Then
  Message = "Alerte !" + Chr(13) + "Débordement"
 Else
  Message = "R A S"
 End If
End Sub

Private Sub Txtcritique_Change()
If CSng(TxtNiveau) >= CSng(TxtProf) Then
  Message = "Alerte !" + Chr(13) + "Puits vide"
 ElseIf CSng(TxtNiveau) < CSng(Txtcritique) Then
  Message = "Alerte !" + Chr(13) + "Débordement"
 Else
  Message = "R A S"
 End If
End Sub

Private Sub TxtNiveau_Change()
If Not IsNumeric(TxtNiveau) Then TxtNiveau = Act
Act = TxtNiveau
If CSng(TxtNiveau) >= CSng(TxtProf) Then
  Message = "Alerte !" + Chr(13) + "Puits vide"
 ElseIf CSng(TxtNiveau) < CSng(Txtcritique) Then
  Message = "Alerte !" + Chr(13) + "Débordement"
```

```
 Else
  Message = "R A S"
 End If
End Sub

Private Sub TxtProf_Change()
 Slider1 = Val(TxtProf.Text)
End Sub
```

13. Listing de «Supan » (fenêtre de surveillance du panneau) :

```
Option Explicit
Private H As Single, W As Single, I As Double, V As Double
Public Imin As Double, Imax As Double, Vmin As Double, Vmax As Double, Pmin As
Double, Pmax As Double
'Const monchemain = "c:\bouhebbal\source\", Pi = 3.141592654
'Private Declare Function ExtFloodFill Lib "gdi32" (ByVal hdc As Long, ByVal x As Long,
ByVal y As Long, ByVal crColor As Long, ByVal wFillType As Long) As Long
'Private Declare Function Arc Lib "gdi32" (ByVal hdc As Long, ByVal X1 As Long, ByVal
Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long, ByVal X3 As Long, ByVal Y3 As
Long, ByVal X4 As Long, ByVal Y4 As Long) As Long
Private Sub Command2_Click()
 Histopan.Show vbModal
End Sub

Private Sub CmdDim_Click()
 FrmDim.Show
End Sub

Private Sub CmdHisto_Click()
 Histopan.Show
End Sub

Private Sub CmdSeuil_Click()
 FrmSeuil.Visible = True
 H = SUPan.Height
 W = SUPan.Width
 FrmSeuil.Top = 0
 FrmSeuil.Left = 0
 Me.Width = FrmSeuil.Width + 100
 Me.Height = FrmSeuil.Height + 400
End Sub

Private Sub Form_Load()
Dim k As Integer
 ' ouvre le fichier de données
 Open MonChemain + "DonPan.dat" For Input As #1
 'initialise les variables
 Imax = Textiinf
```

```
 Imin = Textisup
 Vmax = Textvinf
 Vmin = Textvsup
 Pmax = Textpinf
 Pmin = Textpsup
 Histopan.MSChart1.ColumnCount = 1
 Histopan.MSChart1.AllowDynamicRotation = False
 For k = 1 To Histopan.MSChart1.RowCount
  Histopan.MSChart1.Row = k
  Histopan.MSChart1.Data = 0
 Next
End Sub

Private Sub Form_Unload(Cancel As Integer)
 Close (1)
 Me.Visible = False
 Histopan.Visible = False
End Sub

'Private Sub Picture1_Paint()
' Picture1.Line (LineI.X1, LineI.Y1)-(1000, 1000)
' Label3 = ExtFloodFill(Picture1.hdc, 10&, 10&, CLng(15), 3&)
' Label1 = ExtFloodFill(Picture1.hdc, 10, 10, CLng(&H7FFFFFFF * Rnd), CLng(150000 *
Rnd))
'End Sub
'
Private Sub SeuilAnnuler_Click()
 FrmSeuil.Visible = False
 SUPan.Width = W
 SUPan.Height = H
End Sub

Private Sub SeuilOK_Click()
 FrmSeuil.Visible = False
 SUPan.Width = W
 SUPan.Height = H
 Imax = Textiinf
 Imin = Textisup
 Vmax = Textvinf
 Vmin = Textvsup
 Pmax = Textpinf
 Pmin = Textpsup
End Sub

Private Sub TextV_Click()
 Progres.Value = 100 * (TextV * TextI) / Pm
End Sub

Private Sub Timer1_Timer()
Dim k As Integer, X As Double, Y As String
```

```
'If Tegosw1.Value = False Then Exit Sub
If EOF(1) Then Seek 1, 1
Input #1, V, I
TextV = V
TextI = I
Dim l As Single, t As Single
l = Sqr((LineI.Y2 - LineI.Y1) ^ 2 + (LineI.X2 - LineI.X1) ^ 2)
t = (I / Imax) * Pi / 180# * 0.5 - Pi / 4#
LineI.X2 = LineI.X1 + l * Sin(t)
LineI.Y2 = LineI.Y1 - l * Cos(t)
t = (V / Vmax) * Pi / 180# * 0.5 - Pi / 4#
LineV.X2 = LineV.X1 + l * Sin(t)
LineV.Y2 = LineV.Y1 - l * Cos(t)
 For k = Histopan.MSChart1.RowCount To 2 Step -1
  Histopan.MSChart1.Row = k - 1
  X = Histopan.MSChart1.Data
  Y = Histopan.MSChart1.RowLabel
  Histopan.MSChart1.Row = k
  Histopan.MSChart1.Data = X
  Histopan.MSChart1.RowLabel = Y
 Next
 Histopan.MSChart1.Row = 1
 Histopan.MSChart1.RowLabel = I
 Histopan.MSChart1.Data = I
End Sub
```

14. Listing de «frmDim » (fenêtre du dimensionnement) :

```
Option Explicit
Dim Etape As Integer
Dim I As Integer

Private Sub Command3_Click()
 Me.Hide
End Sub

Private Sub Form_Load()
 Etape = 1
 For I = 1 To 4
  Cadre(I - 1).Top = 0
  Cadre(I - 1).Left = 0
 Next
 Prec.Enabled = False
 Cadre(0).ZOrder 0
 TxtDebit = Debit
 TxtH1 = H1
 TxtH2 = Abs(H2)
 If H2 < 0 Then Immersion.ListIndex = 0 Else Immersion.ListIndex = 1
 TxtPrixPan = LstPrixPAN(PanActuel)
 TxtPdPan = LstPdPAN(PanActuel)
 TxtTranPanKg = 0
```

```
 TxtTranPanm2 = 0
 TxtSpan = LstL(PanActuel) * LstW(PanActuel)
 TxtRendPan = LstRendPan(PanActuel)

 TxtPrixBat = LstPrixBat(BatActuelle)
 TxTTranBat = 0
 TxtRendBat = LstRendBat(BatActuelle)

 TxtConcep = 0
 TxtRealisation = 0
 TxtAchat = 0
 TxtSupplement = 0
 TxtRendConv – 0.95

 TxtPrixPompe = LstPrixPompe(PompeActuelle)
 TxtTranPompe = 0
 TxtRendPompe = LstRendPompe(PompeActuelle)
End Sub

Private Sub Prec_Click()
 If Etape = 2 Then Prec.Enabled = False
 Etape = Etape - 1
 Cadre(Etape - 1).ZOrder 0
 Suiv.Caption – "Suivant >"
End Sub

Private Sub Slide_Change()
 TxtDebit = 100 - Slide
End Sub


Private Sub Suiv_Click()
 Select Case Etape
  Case 1
   If Immersion = "" Then
    MsgBox " Cliquez dans la liste et choisissez SVP."
    Exit Sub
   End If
   If TxtDebit = 0 Then
    MsgBox " Le débit ne doit pas être nul."
    Exit Sub
   End If
   If TxtAlpha < 0 Then
    MsgBox " Le facteur de pertes ne doit pas être négatif."
    Exit Sub
   ElseIf TxtAlpha = 0 Then
    MsgBox " Vous êtes optimiste !" + Chr(13) + " Alpha=0."
   End If
  Case 3
   'dimensionnement
```

```
'1- la Hmt=...
Dim Htm As Single
If Immersion = "Pompe immergée" Then
 Htm = TxtH1 - TxtH2 + TxtAlpha * TxtDebit ^ 2
Else
 Htm = TxtH1 + TxtH2 + TxtAlpha * TxtDebit ^ 2
End If
'2- P(hydrau)=...
Dim Ph As Single, Ro As Single
Ro = 1000 ' eau
Ph = Ro * 9.81 * (TxtDebit / 1000) * Htm 'debit en m^3/s au lieu de l/s

'3- P(motopompe)=...
Dim Pmp As Single
Pmp = Ph / TxtRendPompe

'4- P(onduleur) si actif
Dim Pmpp As Single
Pmpp = Pmp
If FrmSimul.optionconv.Checked Then
 Pmpp = Pmp / TxtRendConv
End If

Dim Tens As Single, Ppan As Single, Wbat As Single
'5-s'il y a des batteries de stockage alors ...
Tens = 12
If FrmSimul.optionbat.Checked Then
 Ppan = Pmpp * (Tens / 24 + (24 - Tens) / 24 / TxtRendBat) / TxtRendPan
 'nombre des panneuax
 LblNPan = Int(Ppan / LstPPAN(PanActuel) + 0.5)
 LblCPan = Format(LblNPan * (TxtPrixPan + TxtTranPanKg * TxtPdPan + TxtTranPanm2
* TxtSpan), "#####0.00")

 'energie, nombre et prix des batteries
 Wbat = Pmpp * (24 - Tens) / TxtRendBat * 3600 ' temps en secondes
 LblNBat = Int(Wbat / LstVnBat(BatActuelle)) / (LstCmaxBat(BatActuelle) -
LstCminBat(BatActuelle) + 0.5)
 LblNBat.Enabled = True
 LblCBat = Format(LblNBat * (TxtPrixBat + TxTTranBat), "#####0.00")
 LblCBat.Enabled = True
 LblRest = Format(Val(TxtPrixPompe) + Val(TxtTranPompe) + Val(TxtConcep) +
Val(TxtRealisation) + Val(TxtAchat) + Val(TxtSupplement), "#####0.00")
 LblTotal = Format(Val(LblRest) + Val(LblCPan) + Val(LblCBat), "#####0.00")
Else ' pas de stockage
 Ppan = Pmpp / TxtRendPan
 'nombre des panneuax
 LblNPan = Int(Ppan / LstPPAN(PanActuel) + 0.5)
 LblCPan = Format(LblNPan * (TxtPrixPan + TxtTranPanKg * TxtPdPan + TxtTranPanm2
* TxtSpan), "#####0.00")
```

```
    'desactiver les objets des batteries
    LblNBat = 0
    LblNBat.Enabled = False
    LblCBat = 0
    LblCBat.Enabled = False

    LblRest = Format(Val(TxtPrixPompe) + Val(TxtTranPompe) + Val(TxtConcep) +
Val(TxtRealisation) + Val(TxtAchat) + Val(TxtSupplement), "#####0.00")
    LblTotal = Format(Val(LblRest) + Val(LblCPan) + Val(LblCBat), "#####0.00")


   End If
  Case 4
   Me.Hide
   Exit Sub
 End Select
 If Etape = 3 Then Suiv.Caption = "Terminé"
 Etape = Etape + 1
 Prec.Enabled = True
 Cadre(Etape - 1).ZOrder 0
End Sub

Private Sub TxtDebit_Change()
 If Not IsNumeric(TxtDebit) Then TxtDebit = 0
 If TxtDebit > 100 Then TxtDebit = 100
 Slide = 100 - TxtDebit
End Sub
```

BIBLIOGRAPHIE

1. K. K. GOPINATHAN : "*A general formula for computing* ", in revue **SOLAR ENERGY**, Vol 4, N° 6, 1988.

2. H. SAHA, P. BASU, S. B. ROY : " *Application of photovoltaic systems – An economic apprisal with reference to* ***India*** ", in revue **SOLAR ENERGY**, Vol 4, N° 6, 1988.

3. W. PALZ : " *L'électricité solaire, une énergie renouvelable*", 1986.

4. R. BARLOW, A. DERRICK : " *Solar pumping for rural developpment : current status and future directions* ", **BWEA/RAL** International Workshop, Rutherford Appleton Laboratory, UK, June 1995.

5. A. HAMIDAT : " *performances et analyse des coûts des sytèmes de pompage photovoltaïque* ", **2ème** Colloque Maghrébin sur l'Hydraulique, 20-21 mai 1997, Zéralda, Algérie.

6. A. HAMIDAT, M. BELHAMEL, A. MALEK, M. T. BOUKADOUM, S. DIAF, : " *Expérimentation et performances d'une pompe mono-cavité progressive* ", in **REVUE DES ENERGIES RENOUVELLABLES**, VOL. 1 (1998), **CDER**, ALGER.

7. Un Groupe d'ingénieurs de l'ASSOCIATION des HYGIÉNISTES et TECHNICIENS MUNICIPAUX (France) : " *Les stations de pompage de l'eau* ", Édition Techniques et documents, Paris, 1986.

8. Chatelain : " *Electricité* ", Tome 4 : les onduleurs, Éditions DUNOD, 1984.