

# Émuler une machine ENIGMA avec Arduino pour celles et ceux qui n'ont pas beaucoup de temps de loisir.

Par Nulentout : mercredi 20 mars 2024.

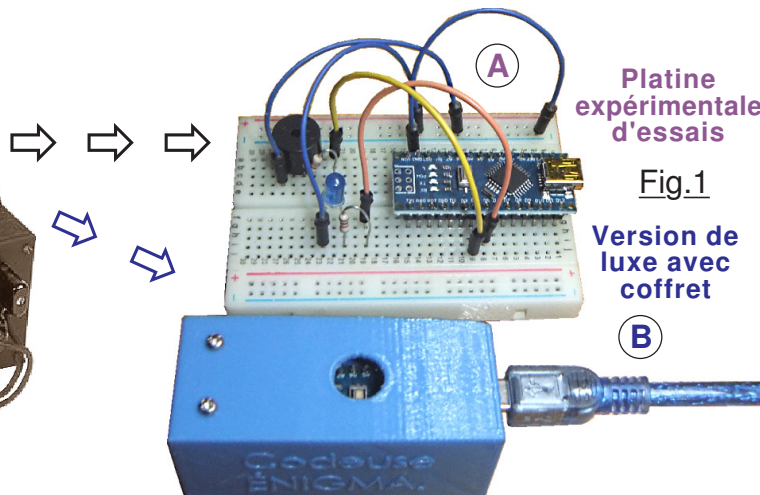
**P**assionnés par la codeuse ÉNIGMA utilisée durant le conflit 1939/1945 vous avez lu de multiples articles sur cette chiffreuse. La possibilité d'en émuler une par utilisation d'une carte Arduino NANO vous tente, surtout pour les sommes assez dérisoires à engager pour un tel petit projet. Toutefois, le didacticiel est prévu pour les programmeurs qui désirent tout savoir sur le fonctionnement interne de cette machine et surtout en aborder la programmation avec méthode.

**C**oncrètement, ce n'est pas que la programmation vous révolue, mais votre vie familiale ne vous octroie pas assez de temps pour vous "cogner" ce trop long didacticiel. Seule la réalisation du petit dispositif vous intéresse et la façon de s'en servir. Ce document vous concerne directement et est fait pour vous. On va se cantonner au strict nécessaire, pour vous permettre de créer la petite simulatrice, avec un minimum d'investissement en temps ... et en argent.



**D**ans ce contexte nous allons totalement éluder le fonctionnement interne de la codeuse. Vous allez vous contenter de réaliser le petit circuit imprimé ou simplement placer une carte Arduino NANO et le petit bruiteur sur une platine à essais. (*Plus une LED.*) Puis vous allez télécharger les trois logiciels en respectant le protocole qui vous guidera dans cette étape. Enfin, des manipulations ordonnées vous expliqueront comment utiliser ce petit dispositif.

**A** votre convenance vous pouvez choisir entre la solution simple de la Fig.1 en **A** qui utilise une plaquette d'expérimentation ou la version de luxe en **B**. Si vous optez pour la version **B** vous trouverez les explications dans le didacticiel en chapitre 3 et chapitre 4 des Pages 6 et 8. Une carte Arduino UNO est totalement compatible avec ce projet, et si vous ne connaissez pas la version NANO vous pouvez consulter sa présentation en Page 5 du didacticiel.



Platine  
expérimentale  
d'essais

Fig.1

Version de  
luxe avec  
coffret

**B**

## 1) Démarrer quand on ne connaît strictement rien d'Arduino.

**E**nvisageons le cas d'une personne totalement naïve au sens noble de ce terme, mais qui par l'entremise d'un ami lointain s'est fait programmer convenablement une carte NANO. Cette connaissance habite à distance et n'a pas forcément le loisir de venir installer le contexte de l'**IDE** sur l'ordinateur. Ce chapitre traite de cette éventualité tout à fait vraisemblable. On part du principe que vous ne saviez même pas qu'existait la ferveur mondiale pour les cartes Arduino qui permettent de programmer de façon très facile le microcontrôleur ATmega328 pour lui faire gérer des petites applications de loisir. (*Encore que dans le domaine professionnel il tient aussi le haut du pavé.*) Le but de ce chapitre consiste à vous faire installer l'**IDE** sur votre ordinateur pour pouvoir y brancher le petit système avec une carte Arduino NANO dans laquelle votre ami a logé le programme et les données en mémoire EEPROM. L'**IDE**, est un environnement qui par l'entremise d'une prise USB de votre ordinateur permet de programmer en langage C++ les cartes électroniques de la famille ARDUINO. Vous ignorerez l'Editeur de texte et le compilateur. La seule fonctionnalité qui vous concerne est le **MONITEUR**. Mais pour en disposer il faut l'**IDE**, raison de ce chapitre.

### ➤ **Installer l'IDE sur l'ordinateur.**

**L**environnement **IDE** est un ensemble de modules informatiques très propre qui n'interfère absolument pas avec Windows. Il est totalement autonome et peut parfaitement être installé sur une simple clef USB. Ceci dit, avec la capacité de stockage des mémoires de masse actuelles, nous n'en sommes plus à 1Go près. Pour ma part, j'utilise depuis des années la version 1.7.9 qui tourne bien sur ma machine gérée par l'ancien Windows VISTA et téléchargée sur :

<http://www.arduino.org/software#ide>

On commence par accepter les cookies. Cliquer sur l'onglet **[Software]**. La version actuelle proposée est la version 2.0.1 qui fonctionne en 64 BITS sur Windows 10. Je ne connais pas cette version n'ayant pas ce système d'exploitation. Quand on explore la page, vers le bas on trouve des versions plus anciennes. Curieusement, dans les références poussiéreuses on trouve les 1.6.*n* et les 1.8.*n* mais pas celle que j'utilise. À vous de

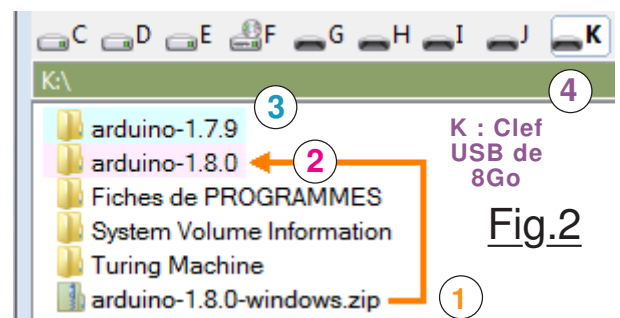



Fig.2

trouver la version qui ressemblera le plus à ce que je décris et surtout qui "tourne" sur votre ordinateur. Pour ce tutoriel j'ai téléchargé la version **1.8.0** pour voir ce que ça donne. Pour l'expérience, comme vous pouvez le vérifier sur la Fig.2 le fichier compressé de 153Mo en 1 a été logé sur une petite clef USB de 8Go. Extrait directement en 2 sur cette petite mémoire de masse extérieure le dossier pèse 410Mo. (*Pour vérifier que l'IDE pouvait fonctionner sur un support extérieur à l'ordinateur, j'avais en 3 recopié le dossier d'utilisation "normale" qui réside sur mon P.C. de bureau.*)

### ➤ **Activer l'IDE sur l'ordinateur.** (Ou sur un disque externe.)

**C'**est de loin la phase la plus compliquée de l'opération. Explorateur de Windows activé, il faut cliquer sans se tromper sur le dossier 2 qui ouvre une liste effrayante de fichiers. Dans cette liste, on cherche avec fébrilité un exécutable intitulé  **arduino.exe**. Courage, vous allez y arriver, on y est presque ! Cliquer nerveusement deux fois rapidement sur le nom de ce fichier étrange. Le truc étonnant de la Fig.3 s'active avec plein plein plein de textes inquiétants en bas à gauche en **A**. Comme ces textes ressemblent à un compte à rebours, je me suis

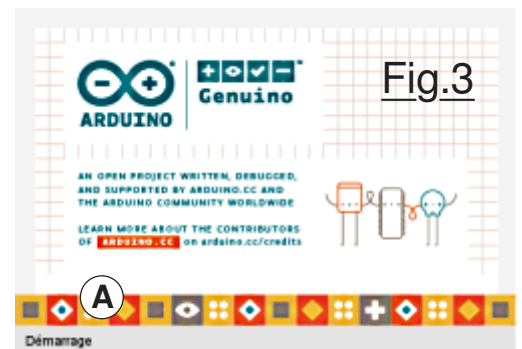


Fig.3

planqué sous le bureau et j'ai attendu une heure. Rien ne s'est passé, à part des crampes dans les jambes. Et,



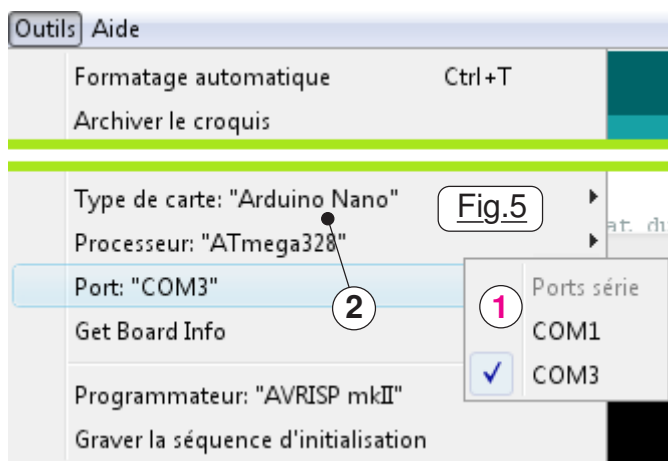
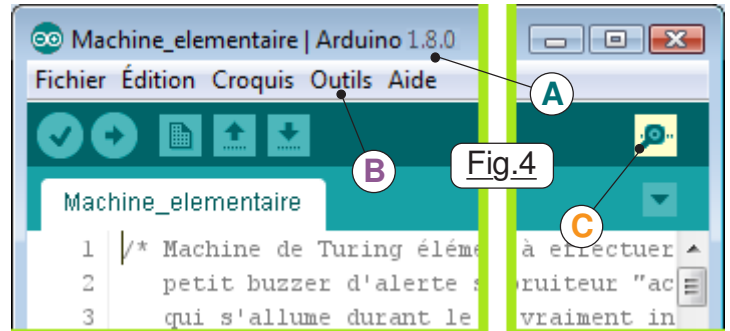
Ben Mômôa j'ai une superbe IDE en tête. Je vais vendre mes Énigma empilées que j'ai trouvé dans le grenier de grand père.

miraculeusement, quand je suis sorti de ma planque, sur l'écran de l'ordinateur il y avait la fenêtre contextuelle de la Fig.4 qui est exactement identique à celle qui s'ouvre quand j'active ma vieille version antédiluvienne 1.7.9 du compilateur C++.

- *GLUPS ... ça tourne !*

### ➤ Activer le Moniteur de l'IDE.

L'historique de ce développement vous n'en avez cure, et vous avez bien raison. La seule chose qui vous importe, c'est d'avoir sur l'écran la fenêtre contextuelle du **Moniteur** de l'**IDE** et que le dialogue s'installe entre la petite machine dans son coffret bleu et l'écran de l'ordinateur. Après avoir effectué les manipulations du chapitre précédent, la fenêtre de l'**IDE** montrée en Fig.4 est active sur l'écran de l'ordinateur. Branchez la petite carte Arduino NANO avec sa mini prise USB sur l'un des ports libres de l'ordinateur. Puis validez l'onglet **[Outils]** en **B**. Si le port est reconnu, on obtient un résultat qui ressemble à celui de la Fig.5 prouvant que la liaison série est correctement établie. Éventuellement si la bonne prise n'a pas été sélectionnée, il suffit de la valider en **1**. Bien qu'en principe ce ne soit pas du tout important si on ne compile pas, on va *par mesure de précaution* indiquer le type de carte raccordé. Dans ce but on



active à nouveau l'onglet **[Outils]** en **B** et avec l'item **Type de carte: "Arduino Nano"** on valide l'option **Arduino Nano**. Il ne reste plus qu'à établir le dialogue machine entre les deux entités informatiques. Il suffit tout simplement de cliquer sur le symbole **C** mis en évidence en jaune sur la Fig.5 en haut à droite. Immédiatement la fenêtre du **Moniteur** de l'**IDE** s'ouvre et affiche le **MENU de BASE**. (Le symbole représente une loupe qui analyse du texte, car le **Moniteur** est l'outil de base pour un programmeur qui sert à déverminer le logiciel qu'il est en train de développer.)

-*HOOOooo NONNnnnon, l'écran affiche n'importe quoi !*

Pas de panique si Arduino contient un programme qui affiche un texte sur le **Moniteur** et que c'est incohérent. C'est normal. Le logiciel qui "tourne" sur le microcontrôleur n'est pas forcément initialisé à la vitesse actuelle de la voie série. Par défaut dans l'**IDE** il peut être totalement différent. Donc, ***pour utiliser mes programmes imposez en bas et à droite la vitesse de 57600baud***. Maintenant il reste à logger dans l'ATmega328 le programme d'exploitation et les données en EEPROM, procédure explicitée dans le chapitre suivant.

## 2) Logger les données dans les cellules "grises" de l'ATmega328.

Probablement qu'une grande majorité des lectrices et des lecteurs Internauts savent parfaitement téléverser des programmes et des données sur une carte Arduino. Toutefois, je vais détailler la procédure pour les Naïfs, (*Naïf étant ici à prendre au sens noble du terme.*) les connaisseurs pouvant passer "en diagonale". Comme cette application impose de gaver des données dans la mémoire EEPROM du microcontrôleur, on doit procéder en deux étapes :

- Remplir l'EEPROM avec des textes employés dans l'interface Homme/Machine ainsi que le codage des organes virtuels d'Énigma.

*Comme cette opération impose de charger dans l'ATmega328 deux petits utilitaires spécifiques, on commence par cette étape, car elle "écrasera" tout logiciel déjà présent.*

- Téléverser le programme d'exploitation.



### > Inscrire les textes dans l'EEPROM.

Quel que soit le support dans lequel vous avez décompressé les fichiers de cette application, aller dans le dossier <Les programmes Arduino> puis :

- 1) Cliquer sur **P00B\_Textes\_en\_EEPROM.ino** qui active l'IDE sur le **sketch** à téléverser.
- 2) Brancher la ligne USB puis cliquer sur l'onglet [**Outils > Port**] **COM<sub>n</sub>** correspondant.
- 3) Cliquer sur l'onglet [**Outils > Type de carte**] et validez **Arduino Nano**.

- 4) Cliquer en **A** sur la flèche colorisée en rouge sur la Fig.6 pour inscrire le programme dans la mémoire du processeur.
- 5) Cliquer en **B** de l'autre côté de la fenêtre contextuelle pour exécuter ce programme et voir le résultat s'afficher dans la fenêtre du **Moniteur** qui vient de s'activer.

Le logiciel précise la version du contenu qui sera inscrit dans

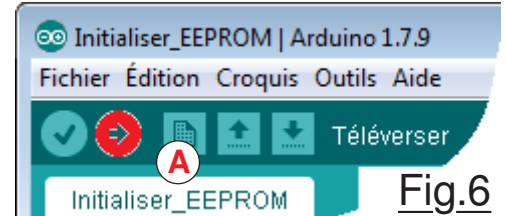


Fig.6

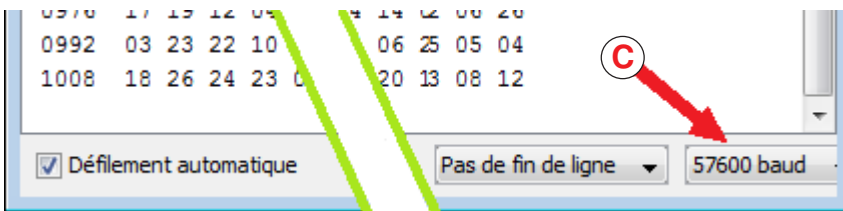


Fig.7



l'EEPROM. Puis, il faut patienter quelques secondes car l'inscription des 1024 Octets exige un petit intervalle de temps. Quand c'est terminé, une foule d'informations se bousculent. *Il est possible, voir probable, que l'ensemble de cet affichage puisse s'avérer incohérent.* C'est tout simplement que la vitesse de dialogue entre le **Moniteur** et Arduino n'est pas synchronisée. Dans ce cas, il suffit d'imposer en **C** de la Fig.7 *une vitesse de 57600 bauds*, car par logiciel je privilégie une cadence de transferts rapide dans les options de l'IDE. Quand cette vitesse est correcte, cliquez une deuxième fois en **B** et les 1024 données sont réinscrites, ce qui n'est pas grave du tout, puis les quatre blocs de mémoire EEPROM sont affichés. Inutile de vous torturer l'esprit pour le moment sur l'analyse de ces données.

### > Inscrire les organes d'Énigma dans l'EEPROM.

Toujours en première étape, on commence par fermer la fenêtre contextuelle de l'IDE pour la rouvrir sur le deuxième outil logiciel **P00C\_Textes\_en\_EEPROM.ino**. On recommence par cliquer en **A** pour téléverser le programme et en **B** pour activer l'écriture dans la mémoire non volatile du microcontrôleur.

### > Inscrire le programme d'exploitation dans le microcontrôleur.

Pour cette deuxième étape, on referme une nouvelle fois la fenêtre de l'IDE pour la réinvoker sur le logiciel d'exploitation du microcontrôleur. Donc, toujours avec l'explorateur de WINDOWS, on visualise le dossier <Les programmes Arduino>.

- 1) Cliquer sur le programme d'exploitation **P13\_EXPLOITER\_Enigma.ino** qui active l'IDE sur le **sketch** à téléverser dans l'étape qui suit.
- 2) Cliquer en **A** sur la flèche colorisée en rouge sur la Fig.6 pour inscrire le programme d'exploitation dans la mémoire du microprocesseur de la carte Arduino NANO.
- 3) Cliquer en **B** de l'autre côté de la fenêtre contextuelle pour exécuter ce programme et voir le résultat dans la fenêtre du **Moniteur** qui affiche la version du logiciel et déroule l'imposant "cadre" du **MENU de Base** de l'Énigma virtuelle.

Normalement l'affichage doit être cohérent puisque la vitesse du Moniteur de l'IDE a été initialisée dans la phase d'initialisation des données dans l'EEPROM. C'est fait, la petite unité est parée pour crypter des centaines de messages "TOP SECRET" ... **CHAMPAGNE** !

Contrairement au didacticiel dans lequel on développe le projet et l'on doit vérifier tout le long la validité des algorithmes utilisés, dans ce document *on fait confiance au programme et on se contente d'apprendre à se servir de toutes ses commandes*. Par convention, une lettre entre cotes, par exemple '**b**', signifie qu'elle doit être immédiatement validée. Une chaîne de caractères encadrée par des guillemets telle que "**bonjour**" sera saisie entièrement avant de la valider.

### 3) Exploiter toutes les ressources du petit simulateur d'Énigma.

Partie la plus agréable, maintenant que l'on dispose d'un petit simulateur de la célèbre codeuse électromagnétique automatique, on va se faire plaisir et voir comment s'en servir. La seule chose à savoir pour les exercices qui vont suivre, c'est que globalement les messages secrets échangés par les états-majors commençaient par un GROUPE d'identification dont les trois premières lettres correspondaient à l'orientation initiale des trois Rotors. Dans l'optique de ce petit tutoriel je fais l'hypothèse que les arcanes de la machines vous sont familiers et que vous savez qu'il faut installer trois Rotors, avec des Indexations internes, un Réflecteur et des FICHES croisées sur un tableau de liaisons filaires. Il est évident que sur notre petite réplique on devra pouvoir effectuer virtuellement ces installations correspondant à l'initialisation de la machine réelle.

#### ➤ Les fonctions de base.

Comme aucune configuration d'initialisation n'a encore été enregistrée en EEPROM, l'affichage est incomplet. Aussi, pour disposer d'un écran cohérent qui pourra être commenté, on va dans un premier temps commencer par quelques commandes non explicitées.

#### MANIPULATIONS :

- 01) Frapper 'd', 'o' et 'o'. (*Attention, valider à chaque lettre car encadré par des cotes.*)
- 02) Proposer 's' suivi de 'o'. (*Dernières commandes non commentées.*)
- 03) Envoyer un '?' : Le Menu de base est affiché dans la fenêtre du Moniteur.



Comme avoir à utiliser la touche [Maj] pour donner les commandes ne va pas dans le sens de la convivialité, l'intégralité des consignes peut se faire en équivalent minuscule. Par exemple pour la commande '?' si [Maj] n'est pas active, la virgule aura le même effet. Dans ce tutoriel, je privilégie les caractères NON [Maj] mais si votre préférence va vers les majuscules, libre à vous.

- 04) Tenter la commande 'q' : Un texte d'erreur vous signale que cette lettre n'est pas valide dans le Menu de base, ce que vous pouviez savoir car elle ne figure pas dans la liste encadrée.
- 05) Imposer l'option 'b' : Maintenant les BIPs sonores d'erreurs sont validés.
- 06) Tester 'u' pour voir : Cette fois le message d'erreur est accompagné d'une alerte sonore.
- 07) Réitérer cette option 'b' : Votre entourage ne sera plus "pollué" par ces bruits intempestifs.
- 08) On se débarrasse de la commande 'p' qui ne concerne que les programmeurs invétérés. Si toutefois vous disposez de dix minutes "à gaspiller", les explications sont en Page 66 du didacticiel.



Le caractère '&' est utilisé chaque fois que l'on veut changer de mode de fonctionnement ou pour anticiper la sortie d'une fonction de saisie de paramètres.

#### ➤ Le cryptage en mode LETTRE à LETTRE.

Avant de "s'embarquer" dans l'expérimentation des nombreuses commandes qui servent à initialiser la configuration de la machine, on va se faire plaisir et explorer les deux modes de chiffrement en commençant par celui où chaque lettre est codée dès que l'on valide sa saisie. L'affichage est alors réalisé ligne à ligne. C'est le mode par défaut lors d'un RESET.

#### MANIPULATIONS : (SUITE)

- 09) Consigner '&' qui fait passer du mode COMMANDE au mode CRYPTAGE. À partir d'ici le texte d'invite réclame une lettre et rien d'autre mis à part le '&' pour quitter le CRYPTAGE.
- 10) Proposer le mot 't', 'e', 's', 't' puis '&' pour revenir au mode COMMANDE.

La Fig.8 présente le contenu du chiffrement avec en vert pastel la lettre frappée au clavier et en bleu

```
Passage au mode CRYPTAGE.
Une lettre -> [A A B] >>> [T] devient [O] ---
Une lettre -> [A A C] >>> [E] devient [L] ---
Une lettre -> [A A D] >>> [S] devient [P] ---
Une lettre -> [A A E] >>> [T] devient [F] ---
Une lettre ->
Retour au mode COMMANDES.
COMMANDE ->
```

Fig.8

clair celle cryptée *compte tenu de la configuration actuelle* de la machine. En jaune est proposé l'équivalent Morse de la lettre une fois chiffrée. Enfin, dans la zone rose on peut observer l'orientation des Rotors sous la petite fenêtre d'Énigma suite à chaque cryptage. Ici seul celui de droite à tourné.

## MANIPULATIONS : (SUITE)

11) Revenir au CRYPTAGE avec le '&'. (Ou le '1' si vous utilisez [Maj].)

12) Tester 'a', 'b', '4', 'c' : Seules les lettres sont acceptées, les autres caractères sont ignorés.



Lorsque l'on est en mode LETTRE à LETTRE le risque est faible, mais en mode TEXTE, comme l'on saisit des phrases entières, par habitude on va inexorablement frapper des accentués, des espaces, et utiliser le point final. Sur une machine réelle, ce risque n'existe pas puisque le clavier ne couvre que les lettres de l'alphabet. Aussi, le programme d'exploitation **P13** "pardonne". Les accentués sont acceptés et convertis en leur équivalent banal. Sont acceptés 'à', 'â', 'è', 'é', 'ê', 'ë', 'î', 'ï', 'ô', 'ç', 'û', 'ü', '.' et l'ESPACE. (Le point et l'espace ne sont pas acceptés en mode LETTRE à LETTRE mais uniquement en TEXTE.)



13) Proposer 'à', 'â', 'è', 'é', 'ê', 'ë', 'î', 'ï', 'ô', 'ç', 'û' et 'ü' : C'est surtout lorsque nous serons en mode TEXTE que ces translations seront appréciables.

14) Tester le point final et l'espace : Donc pas en mode LETTRE à LETTRE.

15) Revenir aux COMMANDES avec '&'. Puis frapper un 'm' : À partir d'ici le Morse sa affiché avec des points et des traits et également en signaux sonores. Ce mode est suspendu par défaut.

16) Imposer 'b' pour suspendre le bruiteur sur erreur ou action "risquée".

17) Passer en CRYPTAGE avec '&' et indiquer quelques lettres : On constate que le Morse sonore est actif même si avec 'b' on a suspendu les BIPs. Ce sont deux commandes indépendantes.

18) Revenir au mode commande avec '&' suivi de ',' pour réafficher le **Menu de base**. On peut y observer l'effet que produit la lettre 'v' à frapper ici pour la tester.

19) Repasser en codage avec '&' puis tester quelques lettres. La cadence est bien plus rapide et ralentit moins le programme. La cadence lente est celle par défaut sur RESET.

### ➤ Structure d'un message secret en 1939 / 1945.

Pendant ce conflit durant lequel la chiffreuse Énigma s'est illustrée pour plusieurs raisons, les protocoles de transmission d'un message étaient fonction des divers services des armées. Toutefois, globalement ils étaient organisés comme montré sur la Fig.9 avec un formatage basé sur des groupes de cinq lettres séparés par des espaces. (Les espaces n'étaient que des silences calibrés durant la transmission Morse on s'en doute.)

Fig.9

AAA	BBB	ONJ	OUR								
BDZGO	WCXLT	KSBTM	CDLPB	MUQOF	XYHCX	TGYJF	LINHN	XSHIU	NTHEO	RXPQP	KOVHC
BUBTZ	SZS00	STGOT	FSODB	BZZLX	LCYZX	IFGWF	DZEEQ	IBMGF	JBWZF	CKPFM	GBXQC
IVIBB	RNCOC	JUVYD	KMVJP	FMDRM	TGLWF	OZLXG	JEYYQ	PVPBW	NCKVK	LZTCB	DLDC
SNRCO	OVPTG	BVBBI	SGJSO	YHDEN	CTNUU	KCUGH	REVWB	DJCTQ	XXOGL	EBZMD	BRZOS
MUQOF	XYHCX	TGYJF	LIN								

Les groupes de cinq lettres ici en jaune constituent le contenu proprement dit du message secret. Compte tenu de sa longueur qui peut être quelconque, le dernier paquet peut contenir moins de cinq lettres. Bien entendu, ce formatage n'est pas issu de la chiffreuse, c'était les opérateurs radios qui respectaient cette présentation qui était spécifique à chaque état-major. Avant le contenu de message, la transmission débutait par le **GROUPE d'identification** ici encadré en violet et contenant douze lettres regroupées par trois. D'une façon générale les trois premières constituaient l'orientation à imposer aux **Rotors** sous la petite fenêtre de la machine. Ici ils sont en rose. (Dans la pratique ces trois lettres étaient différentes et changeaient tous les jours.) Enfin, la zone verte est le complément du **GROUPE d'identification** sachant que pour tous les services d'une arme, ce protocole était commun et indiqué dans des tables ultra confidentielles confiées à l'officier de rang le plus élevé. Pour pouvoir effectuer quelques exercices particuliers, il nous faut encore effectuer quelques petites commandes qui à ce stade du tutoriel ne seront pas commentées.

## MANIPULATIONS : (SUITE)

20) Revenir en mode commande avec '&'.

21) Sans vous poser de question frapper "eg". (ATTENTION : Entre guillemets)

22) Frapper la chaîne "BBBonjour" où un **texte quelconque de neuf lettres**.

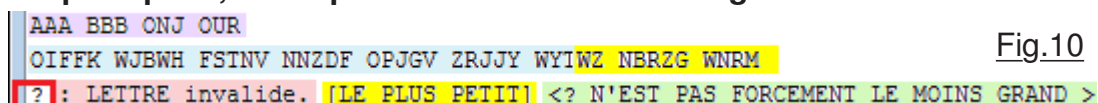
23) Imposer 's' suivi de 'o' et on peut passer à la suite des manipulations.

### ➤ Le cryptage en mode TEXTE.

A voir à valider à chaque lettre soumise à la machine virtuelle est assez laborieux. Le mode LETTRE à LETTRE est bien agréable pour avoir les équivalents Morses, et pour pouvoir par exemple observer le phénomène de **Double Pas** abordé en Page 53 du didacticiel. En revanche, si l'on désire coder des messages relativement longs, il sera bien plus convivial de passer en **mode TEXTE**, c'est à dire pouvoir indiquer toute une phrase sans avoir à valider.

#### MANIPULATIONS : (SUITE)

- 24) Pour la forme "repartir à zéro" avec un RESET : La première action du programme consiste à afficher le **Menu de base** et à lister la configuration initiale de la machine. (*Nous y reviendrons.*)
- 25) Si le bruiteur ne dérange pas votre entourage l'activer avec 'b'.
- 26) Passer en mode TEXTE avec 't'. Le programme confirme cette option et surtout nous précise que le texte proposé ne devra pas dépasser le 60 caractères. C'est une limite imposée par le **Moniteur** de l'**IDE** dont la mémoire tampon est limitée à 63 caractères et surveillé par **P13**.
- 27) Engager le cryptage avec '&'. Noter qu'il n'y a plus le texte d'invite qui précisait "**Une lettre ->**". Le but consiste à ne pas encombrer le message restitué au chiffrement et qui sera présenté en respectant les protocoles pratiqués dans les armées allemandes en 1939/1945.
- 28) Soumettre le texte "**Tout va bien dans le meilleur des mondes**" et valider : Le logiciel commence par afficher le **GROUPE d'identification**, (*Voir descriptif dans le chapitre précédent.*) puis formate le contenu du message en groupes de lettres.
- 29) Tester "**Le plus petit, n'est pas forcément le moins grand.**".



AAA BBB ONJ OUR  
OIFFK WJBWH FSTNV NNZDF OPJGV ZRJY WYIWZ NBRZG WNRM  
[?]: LETTRE invalide. [LE PLUS PETIT] <? N'EST PAS FORCLEMENT LE MOINS GRAND >

Fig.10

#### Quelques remarques s'imposent.

- Dès qu'un caractère interdit est rencontré le chiffrement s'arrête. Pas le mode CRYPTAGE, car on pourra saisir autant de texte que désirés qui seront ajoutés avec continuité du formatage.
- Le fautive est indiqué en premier et repéré dans le cadre rouge.
- On observe que les caractères valides repérés en jaune sur la Fig.10 ont été ajoutés à ceux en bleu clair de la phrase précédente. Dans la ligne du compte-rendu, le texte qui a été accepté et codé est visualisé entre crochets. Le texte qui a été ignoré et qui commence par le caractère incorrect est affiché entre '<' et '>' car il a été effacé dans la fenêtre de saisie du **Moniteur**.
- La virgule interdite a été convertie en '?', opération propre au mode commande.

### ➤ Une aide bien commode : Le Copier / Coller.

Particularité de la fenêtre contextuelle du **Moniteur** : On peut **Copier** dans son champs d'affichage et **Coller** dans sa fenêtre de saisie. Du reste c'est Windows qui gère ce type d'élément. On peut **Copier** et **Coller** dans n'importe quelle fenêtre ouverte y compris échanger facilement des données entre applications différentes de Windows. Expérimentons cette facilité.

#### MANIPULATIONS : (SUITE)

- 30) Dans la fenêtre d'affichage du **Moniteur** **Copier** le texte entre '<' et '>'. (*Sans le '?'*)
- 31) **Coller** ce texte dans la fenêtre de saisie du **Moniteur**.
- 32) Nouvelle erreur car l'apostrophe est illégal.
- 33) **Copier** à nouveau le texte moins l'apostrophe. Ouf, nous avons enfin le codage complet.
- 34) Provoquer un RESET pour recommencer avec un affichage propre.
- 35) Commandes 't' puis '&' pour reprendre un nouveau cryptage en mode TEXTE.
- 36) **Copier** dans cette ligne le texte "**il ne faut pas que ça dégénère. Fin du texte**".
- 37) Le **Coller** dans la fenêtre de saisie du **Moniteur**. (*Le 'ç' et les accentués sont acceptés.*)
- 38) Recommencer la consigne n°37 six fois par exemple.

Tant que l'on n'utilisera pas le caractère de fuite '&' pour sortir du mode et revenir en **COMMANDES**, et qu'il n'y aura pas de caractère illégal, le formatage se poursuivra proprement. Il vous sera tout à fait possible de copier ce texte pour le coller dans tout autre application de Windows. C'est exactement ce que je fais pour rédiger mes didacticiels.



## MANIPULATIONS : (SUITE)

- 39) Soumettre la chaîne de caractères suivante en la *Copiant* : "**Nous allons soumettre à la machine une très longue chaîne de caractères qui volontairement va allègrement dépasser les quarante caractères autorisés par le Moniteur.**"
- 40) Proposer la chaîne de caractères toujours par *Copie* : "**caractères qui volontairement va allègrement dépasser les quarante caractères autorisés par le Moniteur.**" : Tout n'est pas encore accepté car trop long.

*Plusieurs remarques s'imposent.* Considérons dans ce but la Fig.11 qui reproduit en partie l'affichage du Moniteur. Dans la zone en violet on retrouve le GROUPE d'identification. Puis lorsque l'on saisit la première chaîne des caractères trop longue, le programme nous avertit avec le texte de l'encadré jaune et orange, éventuellement suivi d'un BIP sonore si l'option 'b' est activée. Pour que vous puissiez les compter ; dans le texte les caractères sont coloriés par "paquets" de dix. Naturellement, les espaces sont comptabilisés. Du coup, la chaîne validée et coloriée en vert pastel ne compte que cinquante caractères dans le formatage car les espaces sont ignorés. Pour que l'opérateur puisse savoir quelle est la fin du texte qui n'a pas été codée, les caractères pris en compte sont encadrés dans la zone bleue par les délimiteurs '<' et '>'.

### ➤ Technique pour transmettre un long message secret.

Bien que pour des raisons de sécurité les opérateurs radio avaient pour consignes formelles de ne pas envoyer des messages trop long, mais de les scinder en plusieurs transmissions indépendantes en changeant éventuellement à chaque fois le GROUPE d'identification, les tailles des transmissions dépassaient toutefois les soixante caractères. La procédure pour contourner les limites du Moniteur de l'IDE consiste à Copier dans le texte des chaînes inférieures à soixante caractères, et à la soumettre au champs de saisie les une après les autres sans générer de fin de transmission par la commande '&'.

## MANIPULATIONS : (SUITE)

- 41) Provoquer un RESET pour repartir sur une base "propre".



**REMARQUE :** Savez-vous qu'un moyen simple d'engendrer un redémarrage consiste à cliquer sur la "loupe" **B** de la Fig.4 avec l'avantage d'ouvrir à nouveau la fenêtre contextuelle du Moniteur et ainsi d'en effacer toutes les traces des actions précédentes ?

- 42) Commande 'b' si vous désirez le bruiteur.  
43) Classique 't' suivi de '&' pour passer en CRYPTAGE.  
44) Saisir la chaîne "**Nous allons soumettre à la machine une très longue chaîne**".  
45) Continuer avec "**de caractères qui volontairement va allègrement dépasser**".  
46) Compléter avec "**les quarante caractères autorisés par le Moniteur.**".  
47) Revenir au mode COMMANDE avec '&'.

En saisissant des chaînes dont la longueur est notable tout en respectant la limite des soixante caractères on obtient le résultat de la Fig.12 dans lequel en rose est précisée l'orientation des trois Rotors en fin de codage.

```
Passage au mode CRYPTAGE.
AAA BBB ONJ OUR
YIFNO IUQVO JHVRs WEOOB OUBOE UFFXC FYsBZ JBFEP VULNU ZECWV
SALMF FJYEO KBKHD PWUDB WXHEF SQFOJ BWMWV TQDVB GGBGN JQIOM CNPIL OTBEC
TNXXG BHNRC FCBEX LBIIZ P
>>> [B G L]
Retour au mode COMMANDES.
COMMANDE ->
```

GROUPE d'identification

Fig.12

Le message complet comporte 141 lettres chiffrées. Comme ce n'est pas un multiple de cinq, le dernier "paquet" ne contient qu'un seul caractère.



## ➤ Vérifier la réciprocité d'ÉNIGMA.

C onçue avec ce critère impératif, une machine Énigma peut aussi bien crypter que déchiffrer à condition, naturellement, d'avoir une initialisation commune dans les deux opérations. Pour le vérifier, on va se contenter de redonner à notre simulateur le texte qu'il avait encodé. Évidemment, on va récupérer un texte sans ponctuation, sans accents et formaté par groupe de cinq lettres.

### MANIPULATIONS : (SUITE)

48) Avec '&' on quitte le poste de transmission.

49) Avec le même '&' on saute dans l'espace à la station de réception. *(Et surtout on réinitialise la machine aux conditions de la journée. Concrètement, l'opérateur serait obligé avant chaque transmission ou réception, de recalcr les trois rotors en [AAA].)*

**NOTE :** Je me suis contenté de *Copier* dans la fenêtre d'affichage du moniteur l'intégralité du texte crypté. Puis j'ai réalisé des groupements de cinquante caractères en collant cinq groupes et en enlevant les espaces qui dans le **Moniteur de l'IDE** sont comptabilisés.

50) Premier "paquet" :

"YIFNOIUQVOJHVRSWEOBOUBOEUFFXCFYSBZJBFEPVULNUZECWV".

51) Deuxième chaîne :

"HXKQTVCMBSALMFFJYEOKBKHDPUDBWXHEFSQFOJBWMWV".

52) Fin du message : "TQDVBGGGBGNJQIOMCNPILOTBECTNXXGBHNRFCBEXLBIIZP".

```

Passage au mode CRYPTAGE.
AAA BBB ONJ OUR
NOUSA LLONS SOUME TTREA LAMAC HINEU NETRE SLONG UECHA INEDE CARAC TERES
QUIVO LONTA IREME NTVAA LLEGR EMENT DEPAS SERLE SQUAR ANTEC ARACT ERESA
UTORI SESPA RLEMO NITEU R
>>> [B G L]
Retour au mode COMMANDES.
COMMANDE ->
  
```

Fig.13

Nous retrouvons effectivement, Fig.13, le texte d'origine brut dans la zone colorée en vert. Au passage, on observe qu'à la transmission ou au décodage dans les deux cas les trois **Rotors** ont la même orientation finale **[BGL]**. C'est normal puisque dans les deux cas il y a le même nombre de caractères et que le **Brouilleur** entraîne les mécanismes virtuels de façon identique. Pour le vérifier, on peut trouver sur internet le simulateur de la Fig.14 dont l'initialisation par défaut en **2** est identique à celle de notre version Arduino de l'ÉNIGMA. *(Ce n'est pas du tout fortuit, car dans les manipulations non commentées je fais adopter cette initialisation de base.)* Vous pouvez vérifier que sur cette machine totalement indépendante on décrypte strictement pareil en **3** et que les trois **Rotors** en **1** sous la fenêtre de la chiffreuse terminent également en **[BGL]**.

**REMARQUE :** La Fig.14 est constituée d'un montage par copie d'écran de diverses fenêtres récupérées et superposées en **2** et **3** pour ne faire qu'une seule image.

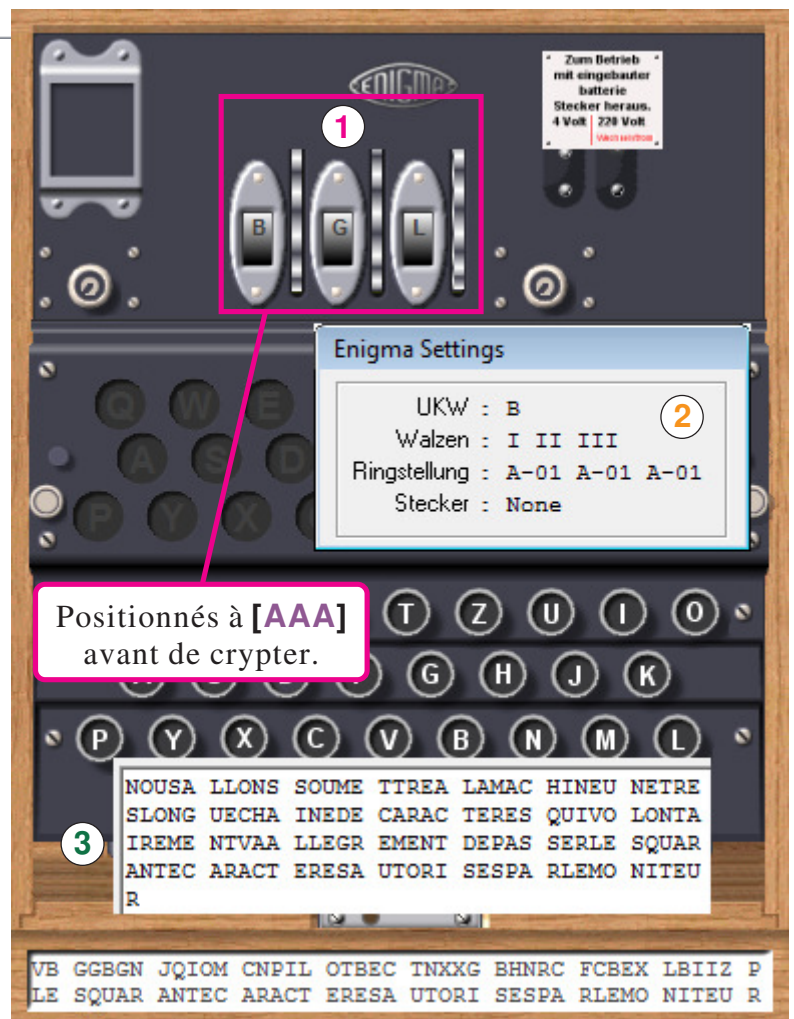


Fig.14

## ➤ Les caractères interdits.

Seules les vingt-six lettres de l'alphabet sont disponibles sur le clavier d'Énigma. Hors, avec notre version informatisée, une foule de caractères peut être soumise par mégarde à la machine virtuelle. Il faut impérativement que le logiciel d'exploitation effectue un filtrage et avertisse l'opérateur en cas d'incident. En mode TEXTE ou en LETTRE à LETTRE le comportement sera forcément un peu différent. Le mieux est encore d'explorer cette facette du projet.

### MANIPULATIONS : (SUITE)

53) Engendrer un RESET comme souvent dans ces expérimentations.

54) Passer en cryptage avec '&'.

54) Proposer successivement 'a', 'b', '5', 'c', '%', 'd', '\$' et '1' ou '&', pour sortir.



En mode **LETTRE à LETTRE**, chaque fois que le programme rencontre un caractère non valide, il se contente de faire un message d'erreur et d'ignorer l'intrus.

55) Imposer le mode TEXTE avec 't' puis revenir au codage avec '&' ou '1'.

56) Tester le texte "**Coucou les amis, il fait beau.**".

Lorsque l'on valide le texte saisi, **P13**

commence par afficher dans la zone violette le **GROUPE d'identification**.

Puis dans la zone verte le début du texte correct est crypté et formaté. Comme

COMMANDE -> [&]

Passage au mode CRYPTAGE.

AAA BBB ONJ OUR

QIFBA PUYKT TXJ

[?] : LETTRE invalide. [COUCOU LES AMIS] <[?] IL FAIT BEAU >

Fig.15

déjà précisé, les caractères sont automatiquement remplacés par leurs équivalents pour faciliter la saisie des commandes. La virgule a donc été remplacée par le '?'. En mode cryptage ce caractère est invalide et provoque la ligne d'erreur qui s'ouvre avec l'affichage du caractère incorrect. Entre crochets dans la zone bleue figure le texte qui a été pris en compte. Enfin, dans la zone verte située entre '<' et '>' se trouve le texte non traité, qui commence par le caractère qui a provoqué l'erreur. Comme il n'y a pas fuite du mode CRYPTAGE avec '&', on peut continuer.

57) Au lieu de frapper à nouveau le texte, il suffit de le **Copier** sans celui qui n'est pas valide et de le **Coller** dans le champ de saisie. Ce complément est alors crypté à la suite.



En mode **TEXTE**, dès que le programme rencontre un caractère non valide, il stoppe le cryptage et génère un BIP d'avertissement. Seule la partie qui précède est chiffrée et affichée avec le formatage de paquets de cinq lettres. Puis ligne suivante le caractère erroné est précisé. S'en suit entre crochets la partie qui a été traitée suivie entre '<' et '>' du texte qui n'a pas été codé, que l'on doit présenter à nouveau à la machine. *(Sans prendre le premier caractère qui n'est pas acceptable.)*



## ➤ Faciliter la saisie des textes soumis à la machine.

Quand un opérateur utilise une vraie ÉNIGMA, son clavier se limite aux vingt-six lettres de l'alphabet, et il lui est impossible de proposer autre chose que des lettres. En ce qui nous concerne, on dispose d'un clavier d'ordinateur avec le risque par habitude de frapper des accentués. Aussi, pour faciliter les manipulations certains caractères spéciaux indiqués dans le tableau de la Fig.16 sont transposés automatiquement. Ces caractères sont convertis en lettres majuscules.

### MANIPULATIONS : (SUITE)

58) Avec '&' on passe en COMMANDES puis 'L' pour choisir le mode LETTRE à LETTRE avantageux pour tester les accentués.

59) Proposer tous les accentués de la Fig.15 en validant à chaque tentative et vérifier l'acceptation de ces caractères.

60) Frapper '&', 't' et '&' pour reprendre le mode TEXTE.

61) Soumettre la chaîne "**Je vais à l'échelle cachée par ça.**".

62) Réinitialiser la transmission avec '&'et '&'.

62) Donner la réciproque "**slqgd xcgwe xteys adyjp yeqge x**".  
On retrouve le texte JE VAIS A L'ECELLE CACHEE PAR CA.

Apostrophe ignorée.	
à devient A	î devient I
â devient A	ï devient I
ä devient A	ô devient O
é devient E	ö devient O
è devient E	ç devient C
ê devient E	û devient U
ë devient E	ü devient U

Fig.16

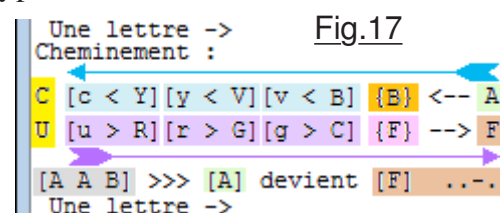
## ➤ Deux commandes réservées aux programmeurs.

Lors de la manipulation 08) "On se débarrasse de la commande 'p'" qui ne concerne que les programmeurs invétérés, j'ai oublié de mentionner deux autres outils '\*' et 'z'. Le premier, consiste à préciser quelle est la lettre sur la bague extérieure où se trouve l'encoche qui fait tourner le **Rotor** s'il est au centre ou à gauche. Cette donnée est purement informationnelle. Le deuxième outil utilisant la commande 'z' et mentionné par (**Zig zag**) dans le **Menu de base** précise à chaque lettre proposée à la codeuse son cheminement et ses transformations en traversant les circuits virtuels de la machine. Cet outils pourra nous être utile si on cherche à réaliser une réplique matérielle d'Énigma. Il ne concerne que les programmeurs et vous pouvez ignorer les manipulations qui suivent.

### MANIPULATIONS : (SUITE)

- 63) En mode COMMANDES frapper successivement '\*', '\*' et '\*' : C'est une bascule de type OUI/NON.
- 64) Imposer 'c' pour faire afficher la configuration de la machine. Cet affichage avec beaucoup d'informations sera commenté dans le chapitre suivant. Observer simplement que le nombre de **FICHES croisées** indiqué entre parenthèses est maintenant suivi des trois lettres d'encocheage des **Rotors** indiqués de la gauche vers la droite et nommés entre crochets.
- 65) Proposer trois fois le 'z'. Ici aussi c'est une commande de type bascule.
- 66) Revenir en cryptage LETTRE à LETTRE avec '&'.
- 67) Faire coder un 'a' par exemple.

On obtient à chaque lettre proposée un descriptif tel que celui de la Fig.17 qui précise le "cheminement" du courant électrique dans les circuits de la machine avec en vert pastel la lettre proposée et en marron clair son cryptage en sortie du tableau des **FICHES croisées**. Dans la zone orange sa première permutation par une **FICHES croisée**. La flèche bleue indique le cheminement à travers **D**, puis **C** et enfin **G** avec en **MAJ**uscule la lettre en entrée et en **MIN**uscule celle en sortie de chaque **Rotor**. (*Droite, Centre et Gauche.*) La zone verticale jaune précise la permutation à travers le **Rélecteur**. La flèche violette traduit le retour depuis le **Rélecteur** en retraversant dans l'ordre **G**, puis **C** et **D**. Enfin dans la zone rose la lettre traverse une deuxième fois une **FICHE croisée**. Dans ce cheminement détaillé à chaque modification, la lettre est devenue dans l'ordre **A > B > v > y > c > U > R > G > C > F > F**. Dans cet affichage la zone grisée indique l'état final de la position des **Rotors** sous la petite fenêtre de la machine.



### 4) Initialiser la configuration d'Énigma.

Cette facette d'utilisation de la machine a incité à créer de nombreuses commandes pour faciliter au maximum cette opération cruciale dans l'exploitation de la chiffreuse. L'expérience montre que configurer entièrement la machine implique l'initialisation de nombreux paramètres, et souvent nous n'avons envie que d'en modifier quelques-uns. Aussi, c'est l'un des aspects de l'utilisation de notre Énigma personnalisée qui mobilise le plus grand nombre de commandes.

## ➤ L'affichage de la configuration sur notre chiffreuse personnelle.

Sous sa forme la plus générale, la configuration de la machine virtuelle sera présenté comme sur la Fig.18 où l'on peut noter l'intégralité de la configuration d'une Énigma. Dans la zone bleue clair sont indiqués les **Rotors** mis en place dans le **Brouilleur**. C'est dans la zone marron pastel qu'est précisé quel est le **Rélecteur** employé. Dans la zone verte est précisée l'**Indexation**

interne pour chaque **Rotor**. Enfin, dans la zone rose on trouve l'orientation à imposer aux **Rotors** sous la petite fenêtre de la machine *avant chaque cryptage*. Ensuite dans la zone jaune est précisé le nombre de lignes filaires installées sur le tableau des **FICHES croisées**. Ici on respecte les protocoles de l'époque, car globalement les machines étaient pourvues de 10 cordons. Noter au

```
>>> CONFIGURATION <<<
ROTOR de Gauche = III Index de Gauche = 12 Orientation = Q
ROTOR du Centre = I Index du Centre = 23 Orientation = Y
ROTOR de Droite = IV Index de Droite = 5 Orientation = W
REFLECTEUR = B (10 FICHES.)
FICHES n°01 -> [A et T] FICHES n°02 -> [B et D]
FICHES n°03 -> [C et X] FICHES n°04 -> [E et R]
FICHES n°05 -> [F et G] FICHES n°06 -> [H et L]
FICHES n°07 -> [J et K] FICHES n°08 -> [O et P]
FICHES n°09 -> [Q et S] FICHES n°10 -> [U et Y]
GROUPE d'identification = QYW WXC ERT HJK
```

Fig.18



passage qu'ici la liste des **FICHES croisées** est ordonnée sur leurs premières lettres classées par ordre alphabétique et repérées en orange. L'affichage se termine par le **GROUPE d'identification** qui contient douze lettres dont les trois premières correspondent à l'**orientation que doivent présenter les Rotors avant de commencer du cryptage ou du décodage**.

➤ **Initialiser intégralement Énigma.**

Lorsque l'on déclenche la commande 'i' le programme nous oblige à préciser l'intégralité des éléments de la configuration matérielle de la machine c'est à dire l'organisation interne du **Brouilleur**, la combinatoire des **FICHES croisées** ainsi que l'orientation de départ des **Rotors**. Comme les trois premiers caractères du **GROUPE d'identification** définissent l'orientation initiale des **Rotors**, bien que les neuf autres caractères ne constituent pas un état de la machine mais uniquement un protocole des armées, **P14** nous en demande également la fin. Ainsi le formatage d'un message crypté en mode TEXTE sera automatiquement précédé du **GROUPE d'identification**.

La commande d'initialisation 'i' impose à l'opérateur de fournir l'intégralité des éléments de configuration matérielle sans possibilité de fuite avec '&'. Il faut renseigner tous les paramètres.

**MANIPULATIONS :** (SUITE)

- 68) Pour la forme "repartir à zéro" avec un RESET.
- 69) Frapper un 'i' pour débiter l'initialisation complète de la machine.
  - Comme on peut activer cette commande par inadvertance, le programme demande de confirmer.
- 70) Confirmer avec 'O' : Ici on se trompe (*Volontairement.*) et on répond avec **zéro** au lieu de 'O'.
  - Chaque fois que le logiciel attend une confirmation, tout caractère autre que 'O' sera considéré comme une réponse négative. Donc un refus et la commande sera ignorée.
- 71) Recommencer avec 'i' suivi de 'o'.
  - Le programme va nous demander de la gauche vers la droite la nature, la configuration interne et l'orientation de départ de chaque **Rotor**.
- 72) Proposer dans l'ordre '3', '1', '4', '12', '23', '5', 'q', 'y' et 'w'. (*Exemple quelconque.*)
  - Comme nous le verrons plus avant, **P14** effectue un filtrage complet et toute valeur illogique sera refusée assortie d'un message d'alerte obligeant à recommencer la saisie.
- 73) Pour le **Réflexeur** indiquer 'b' par exemple.
  - Sur le tableau des **FICHES croisées** on peut brancher un nombre quelconque le cordons électriques virtuels entre zéro et treize. Quand on en a défini le nombre désiré, on quitte avec '&'.
- 74) Pour respecter les protocoles de cette époque, on va brancher dix lignes filaires :
  - Imposer "xc", "hl", "po", "jk", "ta", "qs", "db", "uy", "re", "fg" et sortir avec '&'.
  - Ici on donne les liaisons filaires dans un ordre quelconque qui correspondrait aux consignes du tableau de la Fig.47 donné en page 29 du didacticiel.
- 75) Le programme affiche les trois premières lettres du **GROUPE d'identification** qui correspond à l'orientation des **Rotors** déjà indiquée et demande de compléter avec les neuf dernières lettres. Consigner la chaîne "wxcerthjk" par exemple. (*Neuf lettres quelconques.*)
  - L'intégralité de l'initialisation étant achevée, il y a retour automatique au mode **COMMANDES**.
- 76) Proposer 'c' pour faire lister l'état actuel de la machine.

```

ROTOR de Gauche = III Index de Gauche = 12 Orientation = Q
ROTOR du Centre = I   Index du Centre = 23 Orientation = Y
ROTOR de Droite = IV  Index de Droite = 5  Orientation = W
REFLECTEUR = B (10 FICHES.)
FICHES n°01 -> [X et C]   FICHES n°02 -> [H et L]
FICHES n°03 -> [P et O]   FICHES n°04 -> [J et K]
FICHES n°05 -> [T et A]   FICHES n°06 -> [Q et S]
FICHES n°07 -> [D et B]   FICHES n°08 -> [U et Y]
FICHES n°09 -> [R et E]   FICHES n°10 -> [F et G]
GROUPE d'identification = QYW WXC ERT HJK
  
```

Fig.19

Mis à part l'ordre des **FICHES croisées** repérées ici dans la zone jaune qui n'est pas organisé par ordre alphabétique, le résultat montré sur la Fig.19 ressemble furieusement à celui de la Fig.17 les cordons électriques étant dans ce listage précisés dans l'ordre dans lequel ils ont été définis. On y retrouve également les trois

lettres en rose d'orientation du début des **Rotors** en tête du **GROUPE d'identification** repéré en violet sur cette copie d'écran. Quand on désire vérifier la combinatoire des **FICHES croisées**, il s'avère bien utile de les ordonner, raison de la commande 'j'.

## MANIPULATIONS : (SUITE)

77) La commande 'j' ordonne les **FICHES croisées** par ordre alphabétique avec affichage réordonné de chaque couple. Avec 'c' on retrouve intégralement l'affichage de la Fig.17 qui avait été obtenu suite à cette consigne 'j'.



**NOTE :** Quand on sauvegarde une configuration en EEPROM, les **FICHES croisées** y sont ordonnées par ordre alphabétique même si la commande 'j' n'a pas été sollicitée.

### ➤ Corriger une configuration initiale.

Compte tenu du nombre d'informations que l'on doit définir dans le logiciel, il n'est pas rare du tout de se tromper ne serait-ce que sur une lettre. Avoir tout à recommencer serait un peu rébarbatif. Aussi la commande 'e' pour corriger une ou des **Erreurs** sera la bienvenue. Cette consigne exige deux lettres, la deuxième 'r', 'f' ou 'g' définissant l'entité que l'on désire redonner à la machine. La signification de ce complément est précisée dans l'encadré du bas du **Menu de base**.

78) Commencer par "er" puis dans l'ordre '5', '2', '1', '3', '9', '26', 't', 'v' et 'g'.

• En réalité la commande "er" corrige la totalité du **Brouilleur** et demande aussi le **Réflecteur**.

79) Pour tester, cette fois le on insère virtuellement le **Réflecteur 'c'**. Le logiciel fait revenir automatiquement au mode **COMMANDE**, précédé du listage de la nouvelle configuration.

80) Continuer par "ef" pour modifier le tableau des **FICHES croisées**. Par exemple dans cette correction on va se contenter de six fiches "aq", "zs", "ed".

• Vous pouvez remarquer que chaque demande d'une nouvelle ligne est précédée de l'ordre alphabétique des lettres déjà utilisées qui deviennent interdites puisque sur la machine on ne peut matériellement pas brancher deux lignes dans la même prise.

81) Poursuivre avec "rd" : Le programme refuse ce couple car la lettre 'd' est déjà utilisée.

82) Persister avec "r0" : Le zéro est refusé car ce n'est pas une lettre.

83) Proposer "rr" : (*Nom d'une pipe, faites un peu attention !*)

84) Frapper "rf", "tg", "yh" et '&' suivi de 'j' pour ordonner.

85) Passer à "eg" pour indiquer les neuf lettres suivantes dans le **GROUPE d'identification**. Par exemple indiquer "aaabbbccc". (*Tout caractère interdit obligera à redonner les 9 lettres.*)

### ➤ Ajouter, enlever une Fiche croisée.

Imaginons que l'on a installé dix cordons, mais sur l'un d'eux on s'est trompé sur une lettre. Pour ne pas avoir à reprendre la totalité des dix fiches, la procédure consiste à :

- Enlever la fiche erronée, (*Commande 'k' pour Kill.*)
- Ajouter une **Nouvelle fiche**.

## MANIPULATIONS : (SUITE)

86) On suppose qu'à la place de "tg" on désirait "tp", commencer par 'k' suivi de '9'.

• Une référence erronée ignore la saisie et retourne en mode commandes.

87) Reprenons avec 'k' pour la fiche '4'. Le listage montre bien que **[G et T]** n'est plus dans la liste.

88) Requête 'n' complétée par "ts" : Le programme nous a listé les lettres déjà utilisées et refusera tout doublon. On corrige avec "tp" et on termine par '&' avec 'j' et 'c'.

89) Consigner 'f' que l'on confirme avec 'o' : Toutes les **FICHES croisées** sont enlevées.

90) Tenter d'effacer une fiche avec 'k'.

• Rien n'interdit sur une machine **Énigma** de brancher 12 cordons, certaines machines en étaient équipées. Sur la notre c'est également possible.

91) Proposer "ef" puis "az", "er", "ty", "ui", "op", "qs", "df", "gh", "jk", "lm", "wx", "cv", "bn". (*Pour faciliter la saisie les lettres sont prises dans l'ordre du clavier.*) À la saisie de la fiche n°13 il y a retour automatique au mode commande.

92) Tester 'n' pour ajouter une fiche.

Par ces quelques exemples on a passé en revues toutes les commandes qui permettent de changer la configuration de la machine, mis à part la consigne 'd'. On a expérimenté aussi le filtrage effectué par le logiciel pour détecter les erreurs de frappe ou les fautes de "logique". Tous les incidents n'ont pas forcément été rencontrés, l'analyse syntaxique étant effectuée à chaque saisie.

## ➤ Configuration par défaut.

A de très nombreuses reprises au cours de développement, il s'est avéré très utile de pouvoir en une seule commande imposer une configuration initiale de base. Il existe en ligne de nombreux simulateurs d'Enigma qui à leur mise en service présentent une telle configuration, et sur beaucoup on retrouve celle qui sera adoptée par le programme **P13**.

### MANIPULATIONS : (SUITE)

93) En mode COMMANDE proposer 'd' et confirmer par 'o'.

• Quand on confirme, dans un premier temps le programme ne modifie que le **Brouilleur**. Pour débrancher toutes le **FICHES croisées** il faut confirmer une deuxième fois.

94) Refuser avec 'n' : Seul le **Brouilleur** a été conditionné.

95) Recommencer avec 'd', 'o' et 'o' : l'intégralité de la machine est reconfigurée. Par contre le **GROUPE d'identification** n'est pas modifié car il ne fait pas partie intégrante d'Enigma.

### 5) Les autres commandes de notre simulateur d'Enigma.

Dans les chapitres qui précèdent, nous avons passé en revue toutes les commandes d'exploitation de la chiffreuse. Dans ce chapitre on va expérimenter quelques facilités apportées dans l'usage du petit simulateur personnel. En particulier on va tester la possibilité de sauvegarder une configuration initiale dans la mémoire non volatile EEPROM de l'ATmega328. Pour le plaisir et l'immersion, nous allons utiliser la feuille ultra secrète de la Fig.20 distribuée dans les armées pour le mois de Février 1942.

GEHEIM!

SONDER-MASCHINENSCHLUSSEL : FEVRIER 1942

FEBRUAR 42

Tag	UKW	walzenlage			Ringstellung			steckerverbindungen										Kenngruppen			
29	C	III	I	II	17	19	15	BG	CT	EM	FQ	HY	IR	KW	NS	PV	UZ	KMP	MQI	TEG	QUL
28	C	I	II	V	23	25	20	BQ	CV	DG	EI	JO	KP	LY	NW	RT	XZ	JEQ	TLU	LNS	PMP
27	B	I	II	V	07	15	21	BQ	DU	ES	FY	HN	IX	JV	KT	MZ	OW	UXG	IVJ	GDC	URC
26	C	III	I	V	24	15	09	AH	BT	CD	GI	JX	KL	NQ	PV	RZ	SU	CSU	RXA	WOA	QST
25	C	III	II	I	08	10	11	AV	CF	DK	EW	GM	IL	NQ	PU	RT	YZ	ZNP	KJU	OJZ	WWY
24	B	III	V	II	04	17	13	AL	CG	DN	ES	HP	IM	OW	RT	UZ	XY	JDH	MEE	ZEA	WEV
23	C	IV	V	II	05	20	09	AZ	BV	CG	EX	FO	HM	IW	KT	LR	PS	OWS	VTP	UVZ	IQN
22	C	V	I	IV	24	26	20	BT	CK	DI	FG	JO	MR	PZ	QY	SU	VW	YTM	STX	KZA	FLF
21	C	V	II	III	10	01	25	AR	BZ	DQ	FT	GI	HW	KP	LY	MV	UX	BSX	JOJ	CXD	HLN
20	B	I	IV	V	20	01	11	BN	CS	DU	EG	FM	HR	KO	LQ	PX	TY	VYX	WHJ	QBZ	VTF
19	B	V	III	I	12	03		AD	BF	CW	EV	GN	IM	KL	QT	RY	SX	QKV	JPS	WTD	ILA
18	C	V	II	I	19	24	03	AL	BI	CG	EK	FN	MQ	OV	PU	SZ	WX	NTH	UAF	FEK	TXK
17	C	III	IV	II	09	16	09	CL	EW	FY	GR	IT	JU	KQ	MO	NX	PV	KHH	CGM	CXN	SGY
16	C	II	IV	III	18	03	02	BE	CK	DF	GT	HJ	IX	MN	OP	UW	VZ	DGL	DIJ	XII	ETO
15	B	I	V	IV	15	21	05	AH	BY	CQ	DK	EL	GZ	MS	NW	OR	UX	JCP	PYY	KKC	FOF
14	B	II	IV	I	13	25	25	BZ	CQ	DP	ET	FG	HI	JK	MW	RV	UY	VFD	MKP	ZAW	HZI
13	B	I	III	V	07	17	26	BT	DJ	EO	FS	GX	HP	KZ	MQ	UY	VW	WNL	HNB	MJX	LKU
12	B	IV	V	III	26	23	07	BJ	CQ	DI	FM	GO	HR	KW	PY	TU	XZ	DLP	IWQ	GWY	OAD
11	C	I	II	V	08	13	11	AM	BX	CT	EV	FY	GO	IN	JP	KQ	WZ	NNF	DID	KRR	ROA
10	C	V	II	I	20	11	02	AF	BY	CX	DO	ET	GL	JU	NZ	RS	VW	RUO	NRI	JSJ	PRQ
09	B	I	IV	III	21	06	13	AM	BO	CU	EX	GW	IL	JV	PQ	SZ	TY	PMF	XHG	RIX	YBB
08	B	V	I	II	26	14	14	DN	EP	FU	GX	HI	LT	MW	QS	RY	VZ	WWM	QIH	VZM	NLO
07	C	V	II	III	08	23	17	AQ	BU	DF	EJ	GT	HN	IP	KM	WZ	XY	TLW	ZSL	FCF	YGX
06	B	II	I	V	10	12	20	BP	CI	EK	FJ	GW	HQ	LX	MT	SZ	VY	BKL	MPN	DJE	HRO
05	C	IV	V	III	17	10	17	DU	EY	FI	HP	JR	KW	MX	NT	OQ	SZ	HWN	VYH	KEA	YYC
04	C	III	II	IV	21	07	06	AH	BQ	DR	EG	FX	IN	JY	LV	OU	SW	JHF	SDA	WKK	IZB
03	B	III	IV	I	08	22	10	CV	DS	EX	FP	GL	IZ	JW	KT	MU	NR	RBO	JNB	JAT	FPJ
02	C	IV	III	II	03	05	04	AJ	BE	CT	FS	GH	IM	KX	LY	NV	PU	RLD	KUQ	CZC	BED
01	C	II	IV	V	14	01	07	AE	CV	DH	FU	GK	IT	LX	MR	OW	SY	KHD	XTE	ONP	YTU

### MANIPULATIONS : (SUITE)

96) Pour commencer on va créer une configuration correspondant à celle du 12/02/1942 : Frapper 'i' suivi de 'o' et préciser '4', '5', '3', '26', '23', '7', 'd', 'l', 'p', et 'b' pour le **Rélecteur**.

97) Poursuivre avec "bj", "cq", "di", "fm", "go", "hr", "kw", "py", "tu", "xz" et "&" pour entériner.

98) Compléter le **DLP** du **GROUPE d'identification** avec "iwqgwyoad". (Penser à Copier/Coller.)

99) Commande 'c' pour vérifier la configuration.



**REMARQUE :** Les feuilles d'initialisation telles que celles de la Fig.19 sont obtenues par des logiciels gratuits disponibles sur Internet. Celui utilisé ici et trouvé sur :

<https://www.cipharmachinesandcryptology.com/en/codebook.htm>



ordonne les binômes par ordre alphabétique. Il est inutile si l'on a saisi "linéairement" ces couples d'utiliser la commande 'j'.



### ➤ Sauvegarde / Restitution de la configuration.

Pour observer le comportement de ces deux commandes complémentaires, nous allons "tricher" un peu et compléter le tableau des **FICHES croisées** jusqu'à treize cordons filaires virtuels. Le but est de vérifier que seules dix fiches sont sauvegardées correspondant aux protocoles majoritaires imposés à cette époque. On va aussi vérifier qu'avant de sauvegarder en EEPROM le logiciel **P14** ordonne les fiches par ordre alphabétique.

#### MANIPULATIONS : (SUITE)

- 100) Commande '**n**' pour compléter le tableau des **FICHES croisées**.
- 101) Proposer "**ea**", "**nl**" et "**vs**". Le tableau étant saturé il y a sortie automatique du mode.
- 102) Frapper '**s**' suivi de '**o**' pour confirmer la sauvegarde.
- 103) Débrancher la fiche USB pour simuler le rangement du petit bloc bleu dans un tiroir. le but est de montrer que même des mois plus tard vous retrouveriez cette configuration.
- 104) Rebrancher le petit module sur la prise USB de l'ordinateur.



Sur la mise en service ou suite à un RESET le programme **P14** recharge automatiquement la dernière configuration qui a été sauvegardée. Seules dix **FICHES** sont enregistrées.

- 105) Répéter la commande '**d**' suivie de deux fois '**o**'.
- 106) Imposer '**s**' confirmé par '**o**'.
- 107) Tester alors la consigne '**r**' qui impose une confirmation avec '**o**'.

### ➤ Le respect des protocoles de l'époque.

Strictement aucune obligation nous impose de respecter les protocoles en vigueur dans les armées allemandes en 39/45. On peut pour s'immerger dans le contexte désirer s'en inspirer pour rédiger nos "messages secrets". Ces directives sont résumées dans la **Fiche n° 14** du fichier **FICHES A5.pdf**. C'est précisément pour nous aider dans ce travail de transposition que **P13** a été remplacé par **P14\_Respect\_des\_PROTOCOLES.ino** dans lequel '**u**' a été ajoutée et se charge de :

- Remplacer la virgule par 'Y' et le point final par 'X'.
- Remplacer le "ch" par 'Q'.
- Remplacer 'ä', 'ö' et 'ü' respectivement par "AE", "OE" et "UE".
- Supprimer les ESPACES.

**ATTENTION :** La nouvelle fonction ne gère pas :

- Doubler les noms propres.
- Remplacer les chiffres et les nombres par leur équivalents en toutes lettres.

#### MANIPULATIONS : (SUITE)

- 108) Envoyer la commande '**u**' : Le programme attend une chaîne inférieure à 61 caractères.
- 109) Tester avec "**cdefhéeèâîôû ä ö ü.**". (*Penser à Copier/Coller.*)

Dans cet exemple dont la Fig.21 montre le résultat, les accentués "français" dans la zone rose ont simplement été remplacés par leurs équivalents simples. Le 'C' non suivi de 'h' et le 'H' non précédé de 'c' ont été conservés. Le point final est remplacé par 'X' et surtout, les trois lettres avec des trémats ont été affichées sans ce dernier (*Et coloriées en vert.*) et complétées par le 'E'.

Texte original :  
[CDEFHÉÉÉÉAIOU ä ö ü.]  
Texte traduit :  
[CDEFHÉÉÉÉAIOUAE OEU EX]

Fig.21

- 110) Poursuivre par '**u**' et le texte "**aaa1bbb**" : Le BIP sonore attire notre attention. Seul le texte qui précède l'erreur est converti. Comme ici l'erreur est le chiffre '**1**', il est converti en '&' pour faciliter la saisie des commandes. Donc **P14** ne peut pas savoir si dans la réalité c'est un '&' ou un '**1**' qui ont été soumis au logiciel, d'où les deux caractères qui signalent l'intrus.
- 111) Tester avec '**u**' et "**aaa&bbb**" : Même comportement.
- 112) Enfin, proposer une erreur simple comme **aaa456bbb** sans oublier au préalable le '**u**'.

Si un caractère illégal est rencontré, la transposition s'arrête immédiatement, la suite du texte est ignorée. Le caractère fautif est signalé entre crochets. Il ne reste plus qu'à reprendre la suite. Si la ligne était longue, on peut dans le texte original préciser entre crochet, en **Copier** dans l'écran du **Moniteur** la suite non traitée et de la **Coller** ... en ayant éliminé le ou les caractères incorrects.

## ➤ Une dernière commande réservée aux programmeurs.

Rédigeant ce tutoriel, une vermine de programme qui était passée inaperçue lors du développement du projet est apparue et a "ébranlé" la rédaction en ligne de ce document. Le problème étant épineux, j'ai ajouté provisoirement une commande 'x' qui vraiment ne concerne que les passionnés du C++. Aussi, au final je ne l'ai pas supprimée vu qu'il n'y a aucun inconvénient à la laisser en place. Elle pourrait se montrer utile si un jour on tentait de développer une version matérielle de cette codeuse. Du coup, pour mémoire j'ai ajouté l'information "(X)" sur la ligne dans l'encadré du bas du **Menu de base** réservée à "F : ..."

### MANIPULATIONS : (SUITE)

113) Pour commencer tout débrancher avec 'f' confirmé par 'o'.

114) Commande 'x' pour un premier affichage.

115) Continuer avec "ef" suivi de "az", "er", "ty", "ui", "op" et '&' par exemple.

116) Nouvelle commande 'x' pour pouvoir analyser son effet.

La Fig.22 permet d'expliquer le contenu de cet affichage. Dans la zone colorisée en rose sont listées les lettres déjà utilisées par les **FICHES croisées**. Un '1' signifie "utilisée" alors que '0' traduit la disponibilité de la prise virtuelle considérée. Dans la zone verte nous trouvons une représentation du tableau des **FICHES croisées** qui contient à sa gauche les paires affectées, les '@' traduisant des cellules inutilisées. J'insiste sur le fait que cette commande ne concerne que les programmeurs.

Fig.22

COMMANDE -> [X]
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
1 0 0 0 1 0 0 0 1 0 0 0 0 0 1 1 0 1 0 1 1 0 0 0 1 1
A Z E R T Y U I O P @
COMMANDE ->

## ➤ Un dernier petit message à décoder.

Pour se quitter sur une note ludique, je vous propose de décoder **un message ultra secret, la machine étant en configuration par défaut**. Attention, le texte a été passé à la moulinette de la commande 'u' pour respecter les protocoles de l'époque. Vous allez constater qu'au final, extraire un texte en "clair" demandait un effort de transposition de la part de l'opérateur radio.

### MANIPULATIONS : (FIN)

117) Pour imposer la configuration par défaut frapper 'd', 'o' et 'o'.

118) Tiiit tit tiiit tit / tit tiiit tit tit / tiiit tiiit tit tit ... le message de la Fig.23 vient d'arriver.

```

AAA IWQ GWY OAD
CLXNO CPFGLDJDJTG FYWEZ KUJWC JLKNO RYXXX RZWGE JVRKG JOZCO JVIYH SCMHG
MNZY EYLSG ZFLQO HHVMM TVQRT NYWQU QHFJZ BTAEH AEGHB LOKXL PXSOG MBKSD
PBSBZ BSDQV WIDHD VOKDA YJILR CDPLR OJBGT JWITR YQFOV YKJLF ZQORV ALJWJ
KTPBY YTUSV KEBVF OBLMA LXLEP PDSQU RODRD KPIWT WOTXR CUISQ CGPHV LVWOA
YCWNS TPSHQ XDEVY BQPAW PESLD OOEJ JMZBP YVWTV YOXJA PRQQR GMJSX LFKKG
RKHIH VCSET RMASH QKLLV CBYYV EVYBV IIGAR OF
>>> [B O Z]
```

Fig.23

119) Pour le déchiffrer commencer par 't' suivi de '&'. On va décrypter par groupe de 50 lettres :

120) "CLXNOCPFGLDJDJTG FYWEZ KUJWC JLKNO RYXXX RZWGE JVRKG JOZCO".

121) " JVIYHSCMHGMNZY EYLSG ZFLQO HHVMM TVQRT NYWQU QHFJZ BTAEH".

122) "AEGHB LOKXL PXSOG MBKSD PBSBZ BSDQV WIDHD VOKDAY JILR CDPLR".

123) "OJBGT JWITR YQFOV YKJLF ZQORVALJWJ KTPBY YTUSV KEBVF OBLMA".

124) "LXLEP PDSQU RODRD KPIWT WOTXR CUISQ CGPHV LVWOAY CWNSTPSHQ".

125) "XDEVY BQPAW PESLD OOEJ JMZBP YVWTV YOXJA PRQQR GMJSX LFKKG".

126) "RKHIH VCSET RMASH QKLLV CBYYV EVYBV IIGAR OF".

