

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

Envisageons 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 ignorez 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.**

Lenvironnement **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 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.*)

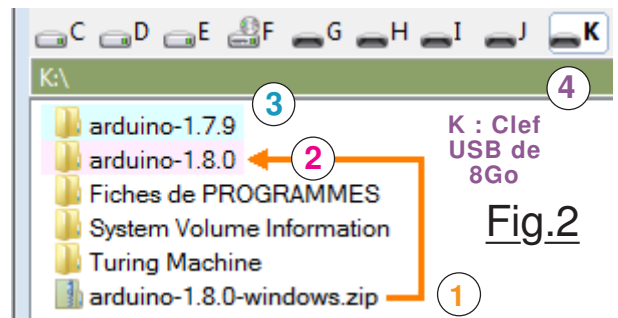



Fig.2

➤ **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

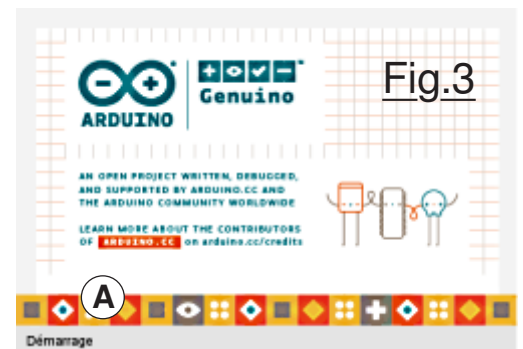


Fig.3

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



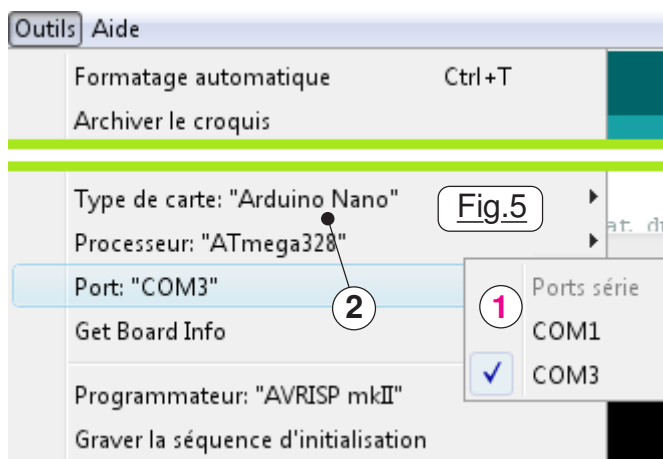
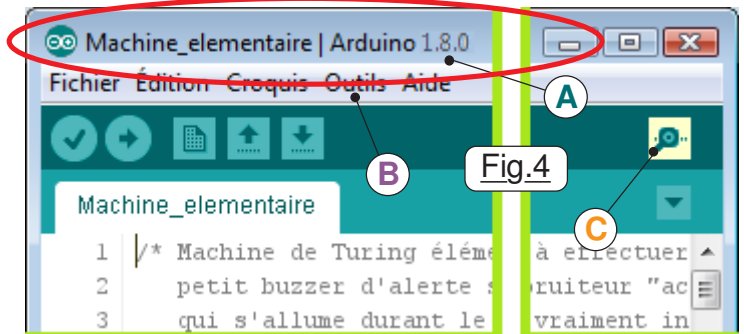
Ben Mômôa j'ai une superbe IDE en tête. Je vais porter à la déchetterie la vieille Énigma de grand père qui ne sert plus à rien !

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 <OUTILS> puis :

- 1) Cliquer sur **P00_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_n** correspondant.
- 3) Cliquer sur l'onglet [**Outils > Type de carte**] et validez **Arduino Nano**.

- 4) Cliquer en **A** sur la flèche colorée 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

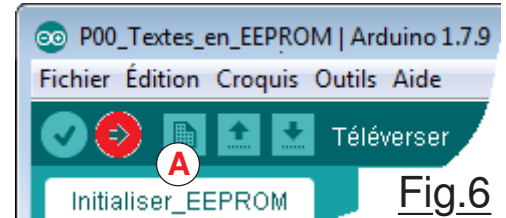


Fig.6

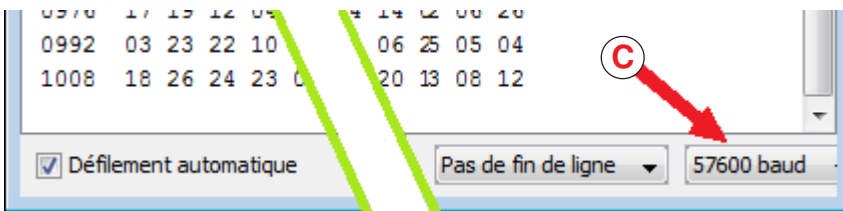


Fig.7



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 l'outil logiciel **P00_Textes_en_EEPROM.ino**. On recommence par cliquer en **A** pour téléverser le programme puis en **B** pour l'activer et ainsi écrire dans la mémoire non volatile du microcontrôleur. *(En fin d'écriture le logiciel liste le contenu de la mémoire EEPROM.)*

➤ 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 colorée 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.* Dans ce but, on imprime **Linéaire.PDF** ou on **XXXXX** qui fournissent un tutoriel pour passer en revue progressivement toutes les commandes de notre réplique d'ÉNIGMA.