



C'est une réalisation amusante basée sur l'ARDUINO UNO. Une fois le sketch mis au point et téléversé, j'ai récupéré le microcontrôleur (ATMEGA 328P) pour le mettre sur un SHIELD réalisé avec EAGLE.

Avant tout, je tiens à remercier F6IDT pour son aide.

Le logiciel (sketch) proprement dit « memoryKeyerV2.0.5.ino » est accompagné des fichiers « AtoM.h, MtoA.h et Canned.h ». L'ensemble devant être placé dans un même répertoire.

Selon la procédure spécifique au soft ARDUINO (nommé IDE) il faudra aussi déclarer la librairie « PS2keyboard », elle apparaîtra alors dans Documents/Arduino/Librairies.

Le shield :

Ce manipulateur permet à partir d'une fiche jack stéréo raccordée sur le connecteur à 3 broches « Manip » d'opérer en double contact (Iambic keyer), ou aussi avec un jack mono en mode pioche ou semi-automatique (voir le schéma ci-dessous).

Un autre connecteur à 4 broches permet de raccorder un clavier PS2.
(Le sketch est prévu pour un clavier AZERTY, il peut être modifié pour un QWERTY).

Les boutons-poussoirs S3 à S6 permettent quant à eux d'émettre 4 messages mémorisés indépendants.

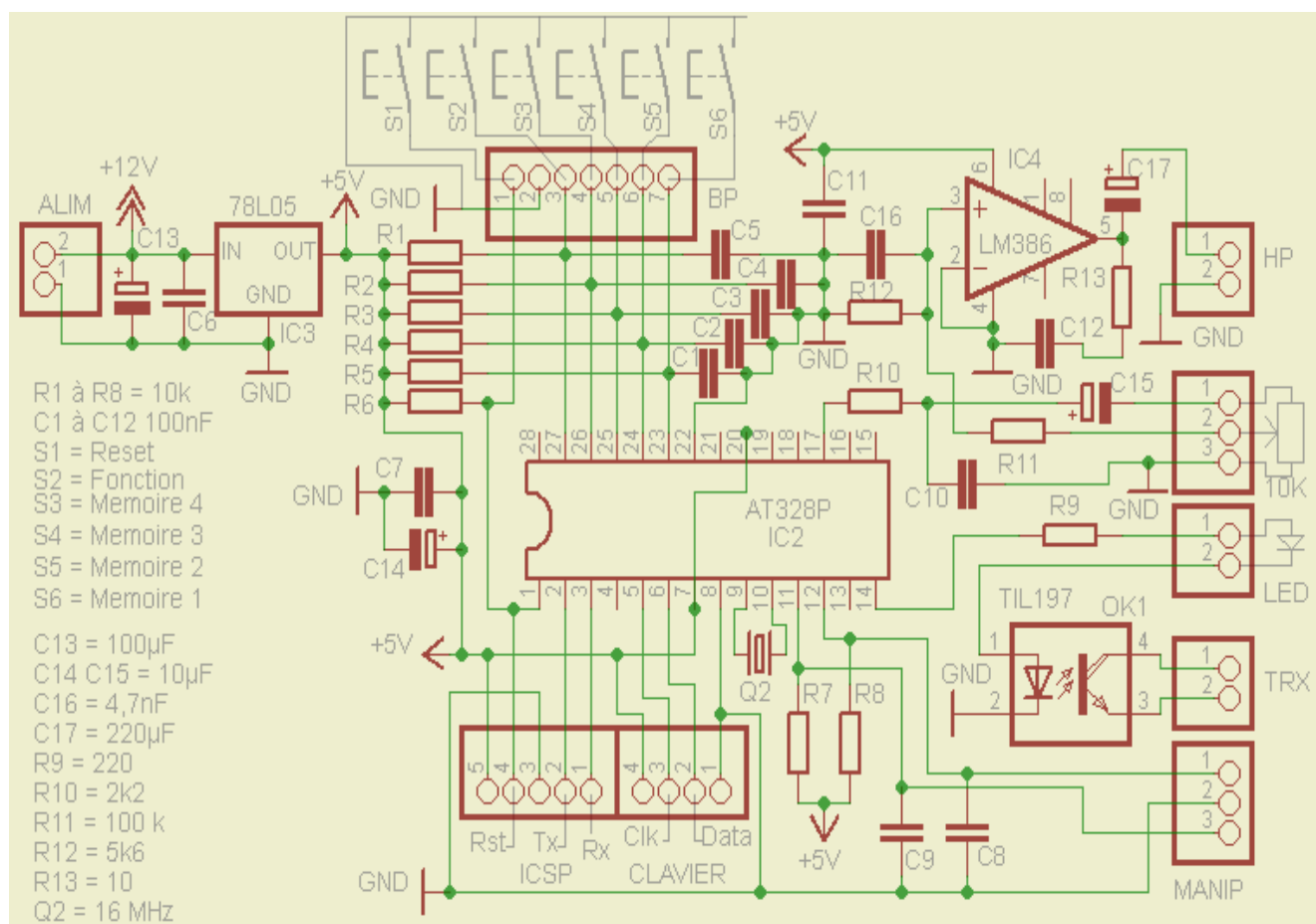
Le connecteur à 2 broches TRX sera raccordé à votre transceiver préféré sur sa fiche de manip (key).

Sur le connecteur à deux broches LED est raccordée une diode led témoin qui clignotera au rythme des messages transmis par les mémoires, le clavier ou encore le manip.

Les tonalités correspondantes au code morse sont audibles par l'intermédiaire du connecteur HP. L'autre connecteur à 3 broches sera raccordé à un potentiomètre de 10 k Ω pour régler le volume sonore. Ce montage permettra aussi de s'entraîner à la télégraphie.

Le schéma :

L'ensemble est alimenté par une tension issue d'un adaptateur secteur 230/9 V 660 mA.
Suit un régulateur 78L05 avec ses condensateurs de découplage C6, C7, C13 et C14.



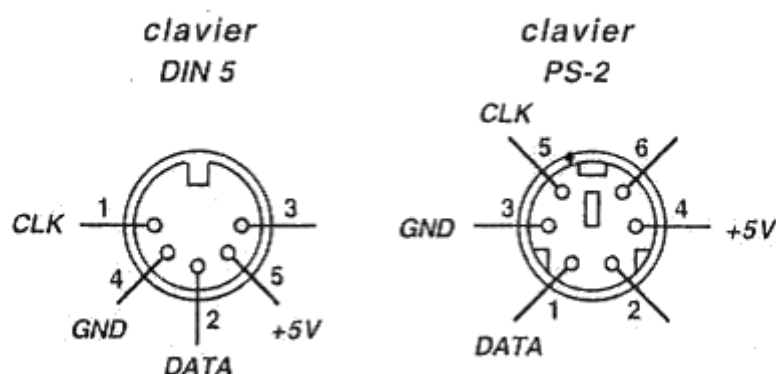
L'ensemble des résistances R1 à R5 fixe le 5 V sur les broches 23 à 27 du microcontrôleur AT328P et les condensateurs C1 à C5 associés d'antirebond des boutons-poussoirs.

L'optocoupleur TIL197 permet l'isolation galvanique avec le transceiver.

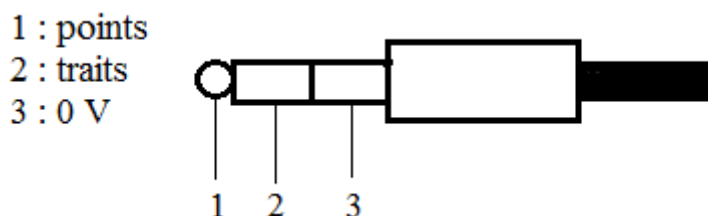
De la broche 17 du microcontrôleur sort le signal audio qui transite par l'intermédiaire de R10 par un potentiomètre de réglage du volume sonore.

Suit un ampli BF (LM386) alimentant un petit haut-parleur de 8 Ω.

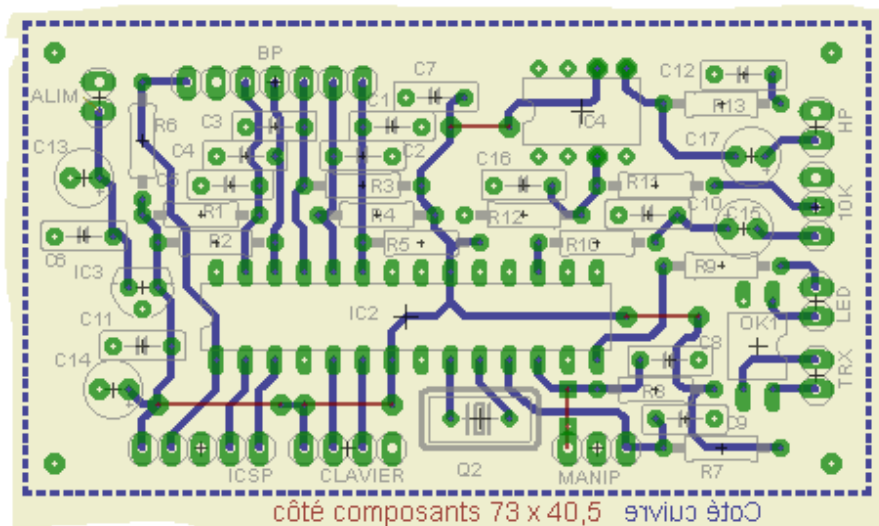
Brochage du connecteur PS2 6 broches et d'un DIN 5 broches :



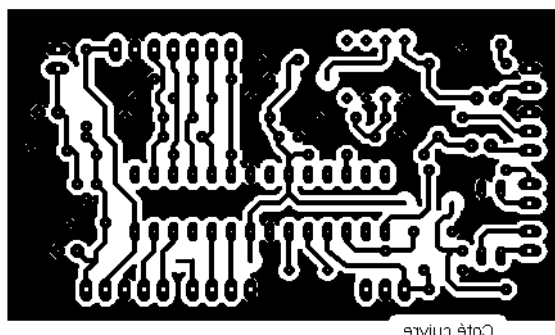
Nota : Il est aussi possible d'utiliser un clavier muni d'une fiche din classique.



Le circuit imprimé :



Il a été réalisé avec EAGLE comme de coutume à F6KKU.



A l'origine, cette réalisation est due à WB8NBS, le sketch a été adapté par F6IDT, notamment en ce qui concerne le clavier PS2.

Nota : Pour utiliser un clavier AZERTY et non QWERTY (cependant plus facile à trouver), la ligne 166 du sketch : `keyboard.begin(DATAPIN, IRQPIN)` est devenue : `keyboard.begin(DATAPIN, IRQPIN, PS2Keymap_French.)`.

Descriptif :

Les caractères à émettre sont placés dans une file asynchrone (voir détail plus loin dans le paragraphe utilisation).

Les quatre mémoires programmables disponibles ont chacune une capacité de 50 caractères. Elles sont programmables à partir du clavier ou du manip.

3 modes d'entraînement/pratique du code morse sont possibles : lettres seules, lettres et chiffres, lettres chiffres et signes de ponctuation.

La vitesse est réglable de 10 à 45 mots par minute (WPM). Ces limites peuvent être modifiées par recompilation.

La fréquence audio est réglable de 100 à 1500 Hz. Ces limites peuvent être modifiées aussi par recompilation.

Les mémoires, la vitesse et la fréquence audio sont stockées en mémoire EEPROM, elles peuvent être facilement remises à leur valeur par défaut.

Un connecteur ICSP 5 broches permet la programmation. Un adaptateur FTDI peut être utilisé pour y raccorder un terminal série. Ainsi les caractères télégraphiques du manip, du clavier seront traduits en ASCII dans le sketch et s'afficheront, idem du clavier (en mode entraînement également) . Je n'ai pas utilisé cette possibilité, préférant une autre solution à paraître dans une future description (le décodeur CW), mais ce connecteur est disponible sur le circuit imprimé.

Il y a six boutons-poussoirs :

Un bouton de réinitialisation pour l'Arduino, S1

Un bouton de fonction, S2.

Quatre boutons pour activer les mémoires individuelles, S3 à S6.

Une pression sur le BP réinitialisation (S1, RESET) provoque différents effets suivant l'état ouvert ou fermé des autres BP. La réinitialisation du processeur prend normalement quelques secondes, une fois terminée, trois bips seront entendus.

- Appuyer sur le bouton de remise à zéro seul pour redémarrer le processus. C'est aussi la seule façon de sortir du mode entraînement.

- Appuyer sur le bouton de remise à zéro tout en maintenant la touche de fonction jusqu'à entendre trois bips pour restaurer les mémoires et la vitesse par défaut.

Les messages par défaut, la vitesse et la tonalité sont définis au moment de la compilation du fichier Canned.h.

Vous devez faire cette action lors du premier chargement du sketch dans un nouveau processeur pour initialiser la mémoire EEPROM.

- Appuyer sur le bouton RESET tout en maintenant l'une des touches de mémoire pour entrer en mode entraînement.

Les caractères seront générés par groupes de cinq.

Mémoire 1 (S6) : lettres seulement.

Mémoire 2 (S5) : lettres et chiffres.

Mémoire 3 (S4) : lettres, chiffres et ponctuation.

- Appuyer sur le bouton de remise à zéro tout en maintenant celui Mémoire 4 (S3) pour accéder au mode de réglage de la tonalité.

Le contact des points augmentera la fréquence, celui des traits la diminuera.

Appuyer sur M4 à nouveau ou saisir \ w au clavier pour écrire la fréquence dans l'EEPROM et sortir.

Le bouton FONCTION offre ces possibilités :

- Le BP FONCTION maintenu enfoncé seul permet de changer la vitesse. Quand il est maintenu, le contact des points augmentera la vitesse, celui des traits la diminuera. La vitesse sera écrite dans l'EEPROM en relâchant ce BP FONCTION.

- FONCTION maintenu enfoncé, en appuyant brièvement sur une touche mémoire permet la programmation de cette mémoire. Une fois la programmation achevée, appuyer à nouveau sur FONCTION pour revenir à la normale. Les messages à programmer peuvent être saisis à partir du clavier PS2, du port série, ou manip.

En appuyant sur une touche mémoire, on transmet le message programmé.

En tapant sur la barre antislash du clavier PS2 « \ » suivie d'un caractère, on entre en mode commande.

Ces commandes sont :

1. \ + ou \ =, augmentation vitesse d'envoi en mots par minute
2. \ -, diminution vitesse d'envoi en mots par minute
3. \ u, augmenter la fréquence audio de 5%
4. \ D, diminuer la fréquence audio de 5%
5. \ w, enregistrer la fréquence audio actuelle et la vitesse (WPM) en mémoire EEPROM
6. \ 1, \ 2, \ 3, \ 4, envoyer un message (comme si un bouton de mémoire avait été enfoncé).

Le tableau touche clavier/morse (AtoM.h) comprend tous les caractères concernant le code Morse. Taper un caractère non pris en charge retournera un « x » et sera ignoré par l'interpréteur morse.

Utilisation du Manipulateur électronique :

Mise sous tension ou remise à zéro :

Lorsque le processeur boote, que ce soit à la mise sous tension ou du bouton poussoir RESET, il initialise le hard et amorce le sketch. Cela peut prendre quelques secondes. Le Manipulateur annoncera alors qu'il est prêt en émettant trois bips.

En maintenant FONCTION enfoncé lors de la réinitialisation du processeur on initialise l'EEPROM à sa valeur par défaut avec les paramètres contenus dans Canned.h, trois bips seront émis lorsque la copie par défaut est terminée, on peut alors libérer FONCTION.

Un défaut d'initialisation de l'EEPROM provoquera des erreurs dans le code de démarrage du sketch avec des effets imprévisibles.

Changement de fréquence audio :

En maintenant la touche MEMOIRE 4 lors de la réinitialisation du processeur, le manipulateur émettra trois bips et vous pourrez libérer MEMOIRE 4. Dans ce mode, le contact des points augmente la fréquence de tonalité, celui des traits la diminue. Quand la tonalité désirée est atteinte, appuyer sur MEMOIRE 4 à nouveau et la fréquence sera enregistrée dans l'EEPROM.

On peut également modifier la fréquence audio à partir du clavier PS2. En saisissant la commande \ u on augmente la fréquence de la tonalité, \ d la diminue. Pour remplacer la valeur par défaut par cette nouvelle tonalité, entrer la commande \ w et elle sera écrite en mémoire.

Lorsque le système écrit une nouvelle fréquence dans l'EEPROM, il l'émet en morse.

Changement de la vitesse d'émission (en mots par minute) :

En tenant le bouton FONCTION enfoncé tout en maintenant le contact des points on augmente la vitesse, tandis qu'avec celui des traits on la diminue.

En libérant FONCTION, le sketch enregistre la nouvelle vitesse par défaut dans l'EEPROM.

Utilisation et enregistrement des messages en mémoire :

Appuyer et relâcher l'un des boutons de mémoire S3 à S6 pour envoyer un message. Les messages sont placés en file d'attente de telle sorte qu'il est possible de presser plus d'un bouton de manière séquentielle. Il est possible de mélanger les messages en mémoire et les entrées au clavier, la logique les enverra dans l'ordre. Toutefois il n'est pas possible d'utiliser un manip (simple ou double contact) et les messages en mémoire de la file d'attente.

Les commandes au clavier \ 1, \ 2, \ 3, \ 4 enverront leur contenu mémoire en file d'attente.

En maintenant FONCTION enfoncé et en appuyant momentanément l'un des quatre boutons de mémoire on entre en mode d'écriture de cette mémoire. En libérant FONCTION le système émet trois bips indiquant qu'il est prêt à enregistrer. Il est alors possible d'entrer des messages à partir du clavier PS2 ou du manip. Appuyer sur FONCTION à nouveau pour enregistrer le nouveau message en EEPROM.

Une autre façon pour saisir le texte en mémoire est d'éditer les messages par défaut dans Canned.h, et de recompiler le sketch bien sûr.

Mode Clé (pioche) :

Un jack à deux conducteurs inséré dans la prise Manip sera détecté au démarrage et le montage entrera en mode clé pioche. Retirer le jack à deux conducteurs remet le montage en mode Iambic (double contact).

Mode apprentissage/entraînement :

Le sketch peut générer aléatoirement du code pour l'entraînement au morse.

Réinitialiser l'appareil tout en maintenant un des trois premiers boutons de mémoire (S4, S5 ou S6). S6 pour seulement des lettres, S5 pour des lettres et des chiffres et S4 pour les lettres, les chiffres et la ponctuation.

Les caractères télégraphiques sont envoyés par groupes de cinq. La séparation entre les groupes peut être ajustée en changeant le paramètre PRACPAUSE dans Canned.h puis en recompilant.

Il faut réinitialiser le processeur pour quitter le mode apprentissage/entraînement (Reset).

Paramètres par défaut du système :

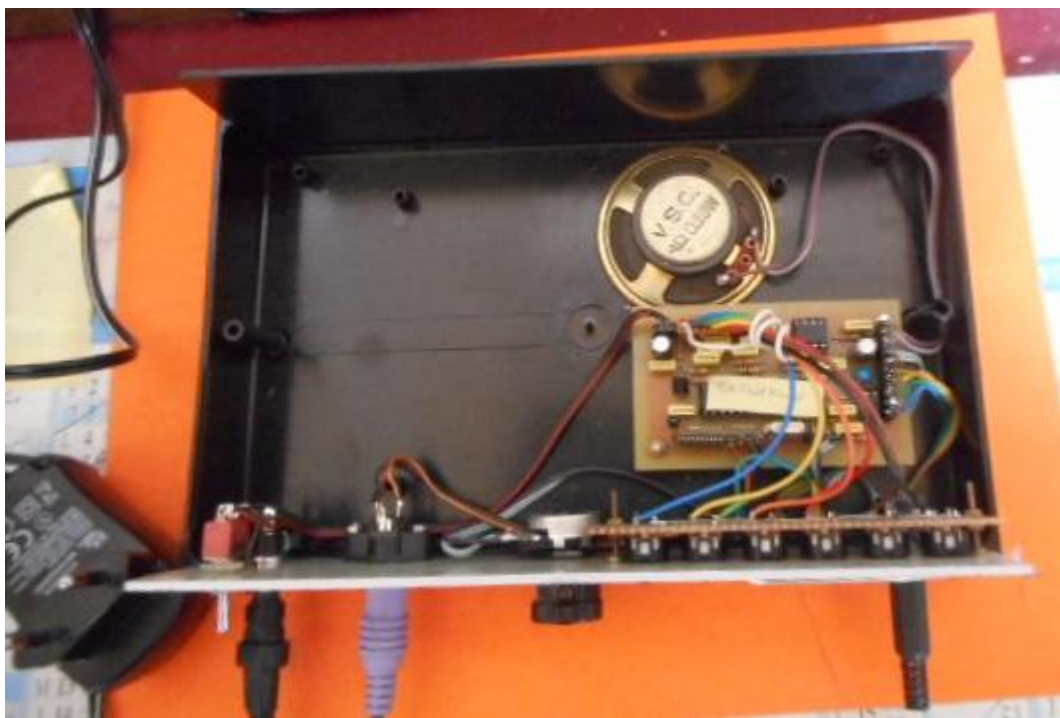
Depuis l'environnement IDE d'Arduino, sur l'ordinateur, il est aisé de modifier de nombreux paramètres par défaut. La plupart sont dans « Canned.h ».

Ces paramètres sont :

1. Les quatre messages en mémoire, jusqu'à 50 caractères chacun.
2. Fréquence audio par défaut.
3. Les fréquences audio maximales et minimales.
4. La valeur par défaut de la vitesse en mots par minute.
5. Ces vitesses maximum et minimum possibles.
6. Les espaces (retard) entre les groupes en mode entraînement/apprentissage.
7. Deux paramètres de réglage de la manipulation :
Réglage du temps d'attente (l'espace) entre les caractères et aussi de celui entre les mots.
8. La longueur de la file d'attente.

D'autres paramètres du programme, tels que l'affectation des broches, peuvent être modifiés d'après le schéma.

Ce descriptif est extrait de : <https://wb8nbs.files.wordpress.com/2015/02/memorykeyer1.pdf> (in English).



Grande boîte, beaucoup d'espace inoccupé mais face avant oblige.

Le soft sera disponible sur le site du RCN-EG (www.RCN-EG.ORG) → Site internet → projets.