**Enigma-Pi**

1. Recherche :
   1. <http://lwh.free.fr/pages/algo/crypto/enigma.htm>
   2. <https://projects.raspberrypi.org/en/projects/octapi-brute-force-enigma/2>
   3. <https://py-enigma.readthedocs.io/en/latest/guide.html#building-your-enigma-machine>
   4. <http://ciphermachines.com/enigma.html>
   5. <http://enigma.louisedade.co.uk/howitworks.html>
   6. <https://www.codesandciphers.org.uk/enigma/plugbd.htm>
   7. <https://www.youtube.com/watch?v=ASfAPOiq_eQ>
   8. <https://media.readthedocs.org/pdf/py-enigma/latest/py-enigma.pdf>
   9. <https://pascalr2blog.wordpress.com/3d-printing/enigma-replica/how-it-works/>
   10. <https://www.thingiverse.com/thing:1813308/#files>
   11. <http://www.instructables.com/id/Make-your-own-Enigma-Replica/>
   12. <https://www.nsa.gov/about/cryptologic-heritage/historical-figures-publications/publications/wwii/assets/files/german_cipher.pdf>
2. Matériel nécessaire :
   * Programme
     + Ordinateur
     + Moniteur
     + Fil HDMI
     + Adapteur HDMI/DVI
     + Raspberry Pi 3
     + Carte SD 32go
     + Clavier
     + Souris
     + Power Supply pour Raspberry Pi
   * Machine physique
     + 1 Arduino Mega
     + 26 Alpha Buttons
     + 26 1/4" Jacks Mono
     + 10 1/4" Plugs Mono
     + 36 Pushbuttons
     + 1 On/Off/On Switch
     + 4 16Segment Orange
     + 4 Injection molded 2-Liter Soda Bottle preform (test tubes)
     + 1 Case Plywood
     + 1 Hinge & Hooks
     + 1 Half-Mortise Lock
     + 1 Perfboard
     + 38 Resistors 470 Ohms
     + 40 Resistors 1K Ohms
     + 7 IRF9Z24N P-Channel MOSFET
     + 1 Piece of Metal & Spray paint

<http://www.instructables.com/id/Make-your-own-Enigma-Replica/>

Marche à suivre

1) Installer la dernière version de Raspbian (système exploitation).

2) Installer le module Py-enigma sur le RaspberryPi.

* sudo apt-get update
* sudo apt-get upgrade
* sudo apt-get install sense-hat
* sudo pip3 install py-enigma

3) Ouvrir IDLE3 ou Py-Charm

* Créer un nouveau fichier que l'on appellera encrypt.py
* Dans ce script : from enigma.machine import EnigmaMachine
* Consulter la feuille : settings sheet pour voir les configurations de la journée (pour la machine).
* Entrez le script (voir page 8-9 de ce document).

**Créer sa propre "Setting sheets"**

from enigma.rotors.rotor import Rotor

from enigma.plugboard import Plugboard

from enigma.machine import EnigmaMachine

r1 = Rotor('my rotor1', 'EKMFLGDQVZNTOWYHXUSPAIBRCJ', ring\_setting=0, stepping='Q')

r2 = Rotor('my rotor2', 'AJDKSIRUXBLHWTMCQGZNPYFVOE', ring\_setting=5, stepping='E')

r3 = Rotor('my rotor3', 'BDFHJLCPRTXVZNYEIWGAKMUSQO', ring\_setting=15, stepping='V')

reflector = Rotor('my reflector', 'FVPJIAOYEDRZXWGCTKUQSBNMHL')

pb = Plugboard.from\_key\_sheet('PO ML IU KJ NH YT GB VF RE DC')

machine = EnigmaMachine([r1, r2, r3], reflector, pb)

NOTE:

Si vous décidez de créer vous-même une « Setting sheet », il faut absolument garder le principe des pairs dans la programmation. C’est-à-dire, si vous brancher le « A » avec le « G », le « G » doit être branché avec le « A » et ainsi de suite.

**Création d’une « setting sheet » pour chaque jour du mois, où le mois contient 31 jours**

*#!/usr/bin/env python***from** enigma.machine **import** EnigmaMachine  
  
  
machine\_01 = EnigmaMachine.from\_key\_sheet(rotors=**'III I II'**,  
 reflector=**'B'**, ring\_settings=**'16 12 02'**,  
 plugboard\_settings=**'AO BY CM DH GI KV LQ RW SZ TU'**) *#EDT UMT ERL LUB*machine\_02 = EnigmaMachine.from\_key\_sheet(rotors=**'II V I'**,  
 reflector=**'B'**, ring\_settings=**'23 09 21'**,  
 plugboard\_settings=**'AP CX DV EU FT GS HI KM LZ NR'**) *#TSW USU CFL VVU*machine\_03 = EnigmaMachine.from\_key\_sheet(rotors=**'V III I'**,  
 reflector=**'B'**, ring\_settings=**'06 05 10'**,  
 plugboard\_settings=**'AV DR EX FY HI JM KZ LQ NS PU'**) *#GCN TBT OOW TNH*machine\_04 = EnigmaMachine.from\_key\_sheet(rotors=**'V IV III'**,  
 reflector=**'B'**, ring\_settings=**'25 15 09'**,  
 plugboard\_settings=**'AF EQ GW HX IO JN KZ MS PR UY'**) *#CFY NRA CNL MVE*machine\_05 = EnigmaMachine.from\_key\_sheet(rotors=**'IV V I'**,  
 reflector=**'B'**, ring\_settings=**'07 06 12'**,  
 plugboard\_settings=**'AI CU DT GS HK JQ LM NV PZ XY'**) *#FYB ITV BOL YID*machine\_06 = EnigmaMachine.from\_key\_sheet(rotors=**'V II IV'**,  
 reflector=**'B'**, ring\_settings=**'20 01 05'**,  
 plugboard\_settings=**'BG CW DI EF JV LZ NY OR PS UX'**) *#KWR MFT ITD GJD*machine\_07 = EnigmaMachine.from\_key\_sheet(rotors=**'I IV V'**,  
 reflector=**'B'**, ring\_settings=**'06 15 10'**,  
 plugboard\_settings=**'AX BP CQ DR FI GY HJ KU MV SZ'**) *#BIF CJM ONT TSM*machine\_08 = EnigmaMachine.from\_key\_sheet(rotors=**'I IV V'**,  
 reflector=**'B'**, ring\_settings=**'02 01 06'**,  
 plugboard\_settings=**'AG CV DH EL FR IT JY MW QU SZ'**) *#RXJ URE ANN ZDD*machine\_09 = EnigmaMachine.from\_key\_sheet(rotors=**'III II I'**,  
 reflector=**'B'**, ring\_settings=**'05 19 12'**,  
 plugboard\_settings=**'BR CT DS EU HW IZ JP LX NO QV'**) *#RLP RWX IQN EGA*machine\_10 = EnigmaMachine.from\_key\_sheet(rotors=**'I V II'**,  
 reflector=**'B'**, ring\_settings=**'22 24 26'**,  
 plugboard\_settings=**'AN BQ DJ EI GU HV KR LP MS XY'**) *#CQQ VEZ YFK NWA*machine\_11 = EnigmaMachine.from\_key\_sheet(rotors=**'III V I'**,  
 reflector=**'B'**, ring\_settings=**'03 03 18'**,  
 plugboard\_settings=**'AP BI CS DU EZ FN HQ KO LM TW'**) *#YEU CZL KLS AJL*machine\_12 = EnigmaMachine.from\_key\_sheet(rotors=**'I II V'**,  
 reflector=**'B'**, ring\_settings=**'14 21 06'**,  
 plugboard\_settings=**'AK BD CL EJ FI GX OR PZ QT VW'**) *#OWC EZZ QXC CAT*machine\_13 = EnigmaMachine.from\_key\_sheet(rotors=**'IV I V'**,  
 reflector=**'B'**, ring\_settings=**'10 06 10'**,  
 plugboard\_settings=**'CZ DR EF GT HS IU LO MV PQ WX'**) *#VXD QEM VOS ECZ*machine\_14 = EnigmaMachine.from\_key\_sheet(rotors=**'I III V'**,  
 reflector=**'B'**, ring\_settings=**'08 14 02'**,  
 plugboard\_settings=**'AL CP DG FY HK JW MS NV QZ TU'**) *#ONT ZAI YNC JPA*machine\_15 = EnigmaMachine.from\_key\_sheet(rotors=**'IV I V'**,  
 reflector=**'B'**, ring\_settings=**'13 03 16'**,  
 plugboard\_settings=**'AE BV FK GO IZ JT LR MX NP WY'**) *#QTQ JIS PCQ QPB*machine\_16 = EnigmaMachine.from\_key\_sheet(rotors=**'III II IV'**,  
 reflector=**'B'**, ring\_settings=**'24 15 14'**,  
 plugboard\_settings=**'CV DJ EK FW GP HS IZ MT QX RY'**) *#GJH QHK GVO BUR*machine\_17 = EnigmaMachine.from\_key\_sheet(rotors=**'II I V'**,  
 reflector=**'B'**, ring\_settings=**'26 09 05'**,  
 plugboard\_settings=**'AX CJ DY EW GP HO IN MS QR UV'**) *#ORH YCH ULT BQY*machine\_18 = EnigmaMachine.from\_key\_sheet(rotors=**'III IV I'**,  
 reflector=**'B'**, ring\_settings=**'08 09 21'**,  
 plugboard\_settings=**'AY BK CS GQ HR JP LO MN UZ WX'**) *#NPU XWU OWU CTI*machine\_19 = EnigmaMachine.from\_key\_sheet(rotors=**'IV I V'**,  
 reflector=**'B'**, ring\_settings=**'02 04 15'**,  
 plugboard\_settings=**'CS DW EF IN JQ KT OX PZ RV UY'**) *#AVD OAG YZI XKV*machine\_20 = EnigmaMachine.from\_key\_sheet(rotors=**'I III V'**,  
 reflector=**'B'**, ring\_settings=**'25 11 10'**,  
 plugboard\_settings=**'AP BE FG IZ KT LN OR QX UW VY'**) *#YMY BFL FDS DBG*machine\_21 = EnigmaMachine.from\_key\_sheet(rotors=**'IV V I'**,  
 reflector=**'B'**, ring\_settings=**'22 01 01'**,  
 plugboard\_settings=**'AF CX DR GN HV JL KZ OW QS UY'**) *#LFE RJG ROD GFU*machine\_22 = EnigmaMachine.from\_key\_sheet(rotors=**'I III V'**,  
 reflector=**'B'**, ring\_settings=**'01 13 01'**,  
 plugboard\_settings=**'AS BC DU GY IK LZ MO NW QX RV'**) *#EGD UUR ODC INE*machine\_23 = EnigmaMachine.from\_key\_sheet(rotors=**'II V I'**,  
 reflector=**'B'**, ring\_settings=**'10 04 12'**,  
 plugboard\_settings=**'BE CK DX FG HI MO NP QU RS VZ'**) *#WDS QXB YYB NDF*machine\_24 = EnigmaMachine.from\_key\_sheet(rotors=**'V I IV'**,  
 reflector=**'B'**, ring\_settings=**'19 02 06'**,  
 plugboard\_settings=**'AL CM DO EP HN IK SZ TU VY WX'**) *#FPC PHY KWY RLH*machine\_25 = EnigmaMachine.from\_key\_sheet(rotors=**'III I IV'**,  
 reflector=**'B'**, ring\_settings=**'06 13 13'**,  
 plugboard\_settings=**'BX CE DR FG HV IN LY MT OS UW'**) *#PWC MPI RAJ VWW*machine\_26 = EnigmaMachine.from\_key\_sheet(rotors=**'IV II III'**,  
 reflector=**'B'**, ring\_settings=**'19 20 20'**,  
 plugboard\_settings=**'AK BX CQ DO EI HS LR PW TZ VY'**) *#SQW JVL MUA LSP*machine\_27 = EnigmaMachine.from\_key\_sheet(rotors=**'IV III II'**,  
 reflector=**'B'**, ring\_settings=**'02 03 15'**,  
 plugboard\_settings=**'AC BP EZ FX GV HK LT MR NU QW'**) *#ZON KOV FCE CDJ*machine\_28 = EnigmaMachine.from\_key\_sheet(rotors=**'IV II III'**,  
 reflector=**'B'**, ring\_settings=**'12 07 08'**,  
 plugboard\_settings=**'BW DZ EY FR GT HJ IV LS NQ OX'**) *#EPK PLF PIR NCY*machine\_29 = EnigmaMachine.from\_key\_sheet(rotors=**'V III I'**,  
 reflector=**'B'**, ring\_settings=**'09 25 21'**,  
 plugboard\_settings=**'AK BU CI DE GQ HM LT OZ RV WY'**) *#AFU WWW TSM UVX*machine\_30 = EnigmaMachine.from\_key\_sheet(rotors=**'II V III'**,  
 reflector=**'B'**, ring\_settings=**'22 11 01'**,  
 plugboard\_settings=**'AL BS EU FR GM IO PY QZ TW VX'**) *#ZUF SIC FDH BKU*machine\_31 = EnigmaMachine.from\_key\_sheet(rotors=**'IV V I'**,  
 reflector=**'B'**, ring\_settings=**'24 17 13'**,  
 plugboard\_settings=**'AQ CT DV EN FW GP IX JS KR LO'**) *#BZA KZO SIK VMO*

**Encrypter et décrypter un message à partir des paramètres d’une journée ci-dessus.**

**from** enigma.machine **import** EnigmaMachine  
**global** machine  
machine = EnigmaMachine.from\_key\_sheet(

rotors=**'IV I V'**,  
reflector=**'B'**,

ring\_settings=**'13 03 16'**,  
plugboard\_settings=**'AE BV FK GO IZ JT LR MX NP WY'**) *#QTQ JIS PCQ QPB***def** encrypter():  
  
 **global** message  
 **global** machine  
  
 *# Initialiser les rotors. Je peux mettre n’importe quoi.* machine.set\_display(**'LOL'**) **print**(**"LOL"**)  
  
 *# Encrypter 3 lettres et c’est le résultat que nous envoyons lors d’un message.* msg\_key = machine.process\_text(**'QTQ'**) **print**(msg\_key)

*#Initialiser la position de la machine avec la clé encryptée.*  
 machine.set\_display(msg\_key)

*#Stocker le message entré dans une variable.*  
 plaintext = message

*#Encrypter le message*  
 ciphertext\_encrypt = machine.process\_text(plaintext)  
 **print**(ciphertext\_encrypt)  
  
**def** decrypter():  
  
  
 **global** message  
 **global** machine  
 *# Initialiser les rotors à la même position que la machine qui a encrypter en entrant les mêmes lettres.* machine.set\_display(**'LOL'**)  
  
 *#Insérer la clé encrypté comme clé du message de decryption* msg\_key = machine.process\_text(**'GFS'**)  
 **print**(msg\_key)  
  
 *# Set the new start position of the Enigma rotors* machine.set\_display(**'GFS'**)

*#Remplacer ça par le message encrypté*  
 ciphertext\_decrypt = **'RHLLQAS'** plaintext = machine.process\_text(ciphertext\_decrypt)  
 **print**(plaintext)  
  
  
*###################### MAIN ###############################***print**()

*#Entrez le message que vous voulez encrypter.*  
message = input(**"Saisir message : "**)

*#Fonction d’encryptage*  
encrypter()  
**print**()

*#Fonction de décryptage.*  
decrypter()  
  
input()

Dans une situation réelle, un opérateur envoi un message encrypter à un autre opérateur de cette façon :

U6Z DE C 1500 **=** 24 **=** LOL GFS **=**

*Clé message encrypté*

BNUGZ **RHLLQAS**

**Ci-dessus, nous avons les informations principales transmises, soit :**

* **Les 3 lettres pour initialiser les rotors(LOL)**
* **La clé d’encryptage (GFS)**
* **Le code (en jaune) permettant de trouver les bons réglages sur la « setting sheet ».**
* **Le message codé (en vert).**

### Par défaut, un X en Enigma signifie un espace. En encryptions, l'espace s'encrypte en en lettre, et en décryptions, il redevient un X pour séparer les mots du message.