

ENIGMA Code simulators

[enigma.pdf](#)

This module is part of the original MMBasic library. It is reproduced here with kind permission of Hugh Buckle and Geoff Graham. Be aware it may reference functionality which has changed or is deprecated in the latest versions of MMBasic.

ENIGMA.BAS

```
'ENIGMA for Maximize
'written by Ray Alger April 2015
'Output file added by Hugh Buckle May 2015
OFileName$="EnigOut.txt"
Dim RD$(10):Dim D$(4):Dim R(4):Dim V(5):Dim RS(4):Dim RP(4)
Dim PB(26):Dim PBD(26):Dim PBS(26):Dim REF(26):Dim K0(8,2)
Dim ROT$(10):Dim REF$(2):Dim FOR0(26):Dim REV0(26):Dim FOR1(26)
Dim REV1(26):Dim FOR2(26):Dim REV2(26):Dim FOR3(26):Dim REV3(26)
'Rotor display data
RD$(1)=" I ":RD$(2)=" II ":RD$(3)=" III":RD$(4)=" IV ":RD$(5)=" V "
RD$(6)=" VI ":RD$(7)=" VII":RD$(8)="VIII":RD$(9)="beta":RD$(10)="gamm"
'rotor knock on data
K0(1,1)=16:K0(2,1)=4:K0(3,1)=21:K0(4,1)=9:K0(5,1)=25
K0(1,2)=16:K0(2,2)=4:K0(3,2)=21:K0(4,2)=9:K0(5,2)=25
K0(6,1)=25:K0(7,1)=25:K0(8,1)=25
K0(6,2)=12:K0(7,2)=12:K0(8,2)=12
'Rotor data
ROT$(1)="EKMFLGDQVZNTOWYHXUSPAIBRCJ" 'I
ROT$(2)="AJDKSIRUXBLHWTMCQGZNPYFVOE" 'II
ROT$(3)="BDFHJLCPRTXVZNYEIWGAKMUSQO" 'III
ROT$(4)="ESOVJPZJAYQUIRHXLNFTGKDCMWB" 'IV
ROT$(5)="VZBRGITYUPSDNHLXAWMJQOFECK" 'V
ROT$(6)="JPGVOUMFYQBENHZRDKASXLICTW" 'VI
ROT$(7)="NZJHGRCXMYSWBOUFAIVLPEKQDT" 'VII
ROT$(8)="FKQHTLXOCBJSPDZRAMENIU YGV" 'VIII
ROT$(9)="LEYJVCNIXWPBQMDRTAKZGFUHS" 'beta
ROT$(10)="FSOKANUERHMBTIYCWLPZXV GJD" 'gamma
'Reflector data
REF$(1)="ENKQAU YWJICOPBLMDXZVFTHRGS" 'Thin B
REF$(2)="RDOBJNTKVEHMLFCWZAXGYIPSUQ" 'Thin C
'Plug board data
PB$ ="QWERTZUIOASDFGHJKPYXC VBNML"
For J=0 To 25:PBD(J)=Asc(Mid$(PB$,J+1,1))-65:Next J
DAY$="C VI I VII XEL ML YE JH WQ"
TRI$="JEX GFE"
CIPHER$="YKDMD SZZPA BFCMJ EZIGN MSROU GPDSO HHSIU JJLSC NBBTY UOTX"
Function MOD26(X)
X=Cint(X) 'fix error
X=26*(X/26-Int(X/26))
```

```
If X<0 Then X=X+26
MOD26=X
End Function
'Screen setup
START: Cls:Print" ***** Maximite M4 ENIGMA *****"
Print@(0,15)"REFLECTOR ->"
Print@(0,30)"ROTORS USED -->"
DP=30
GoSub DRAW
Print@(0,60)"RING SETTINGS ->"
DP=60
GoSub DRAW
Print@(20,120)"ROTORS -->"
DP=120
GoSub DRAW
Line(125,95)-(130,155),,B:Line(175,95)-(180,155),,B
Line(225,95)-(230,155),,B:Line(275,95)-(280,155),,B
For J=1 To 6
  L=J*10
  Line(126,90+L)-(129,90+L):Line(176,90+L)-(179,90+L)
  Line(226,90+L)-(229,90+L):Line(276,90+L)-(279,90+L)
Next J
Line(1,325)-(330,390),,B
Print@(36,385)"PLUG BOARD"
For J=0 To 25:PB(J)=J:Next J 'Init Plug array
For J=0 To 25:PBS(J)=0:Next J 'Init Plug status
GoSub SHOW
Line(1,175)-(331,240),,B
Print@(36,235)"LAMP BOARD"
CODE=26 'no lamp
GoSub LAMP
GoSub REFSET
GoSub ROTSET
GoSub RINGSET
GoSub PLUGSET
GoSub SETROT
'HFBmod 20150509 start - add output file
  gosub OpenOutFile
'HFBmod 20150509 end
'rotor offset arrays
For J=0 To 25
  T=Asc(Mid$(ROT$(R(0)),J+1,1))-65
  U=Asc(Mid$(ROT$(R(1)),J+1,1))-65
  V=Asc(Mid$(ROT$(R(2)),J+1,1))-65
  W=Asc(Mid$(ROT$(R(3)),J+1,1))-65
  FOR0(J)=T-J
  REV0(T)=J-T
  FOR1(J)=U-J
  REV1(U)=J-U
```

```

    FOR2(J)=V-J
    REV2(V)=J-V
    FOR3(J)=W-J
    REV3(W)=J-W
Next J
'knock on data
K03A=K0(R(3),1):K03B=K0(R(3),2)
K02A=K0(R(2),1):K02B=K0(R(2),2)

'refelector data
For J=0 To 25:REF(J)=Asc(Mid$(REF$(RF),J+1,1))-65:Next J
Print@(0,250)"Text IN-"
Print@(0,280)"Text OUT-"
CLRPAD: PL$="":CI$="":CCNT=0:TB=0
Print@(0,265)"
"
Print@(0,295)"
"
Print@(0,410,2)"Enter text, Press[Esc] to change settings      "
ENCRYPT:
TB=TB+1:If TB>60 Then GoTo PADFUL
GoSub LETTER
If K=27 Then GoTo ESCAPE
If K>25 Then GoTo ENCRYPT
'reflector steps
If RP(3)=K03A Or RP(3)=K03B Then GoTo ADV2
If RP(2)=K02A Or RP(2)=K02B Then GoTo ADV1 ' R2 double step
GoTo ADV3
ADV1: RP(1)=RP(1)+1
If RP(1)>25 Then RP(1)=RP(1)-26
ADV2: RP(2)=RP(2)+1
If RP(2)>25 Then RP(2)=RP(2)-26
ADV3: RP(3)= RP(3)+1
If RP(3)>25 Then RP(3)=RP(3)-26
'reflector display
DP=120:D$(1)=" "+Chr$(RP(1)+65)+" "
D$(2)=" "+Chr$(RP(2)+65)+" ":D$(3)=" "+Chr$(RP(3)+65)+" "
GoSub VIEW
'scramble
PL$=PL$+Chr$(K+65)
CODE=PB(K)
X=MOD26(CODE+RP(3)-RS(3))
X=MOD26(CODE+FOR3(X))
CODE=X
X=MOD26(X+RP(2)-RS(2))
X=MOD26(CODE+FOR2(X))
CODE=X
X=MOD26(X+RP(1)-RS(1))
X=MOD26(CODE+FOR1(X))
CODE=X
X=MOD26(X+RP(0)-RS(0))

```

```
X=MOD26(CODE+FOR0(X))
CODE=REF(X) 'Reflect
X=MOD26(CODE+RP(0)-RS(0))
X=MOD26(CODE+REV0(X))
CODE=X
X=MOD26(CODE+RP(1)-RS(1))
X=MOD26(CODE+REV1(X))
CODE=X
X=MOD26(X+RP(2)-RS(2))
X=MOD26(CODE+REV2(X))
CODE=X
X=MOD26(X+RP(3)-RS(3))
X=MOD26(CODE+REV3(X))
CODE=PB(X)
CI$=CI$+Chr$(CODE+65)
'HFBmod 20150509 start - add output file
'Write encrypted code to output file in groups of 5 characters
'New line after 12 character groups
Print #1,Chr$(CODE+65);
I0=I0+1
if I0>4 then ' Space after 5 characters
    I0=0
    IG=IG+1
    If IG>12 then
        IG=0
        Print #1, Chr$(13) ' new line
    else
        Print #1, " ";
    endif
endif
'HFBmod 20150509 end

GoSub LAMP
CODE=26 'no lamp
Pause 500
GoSub LAMP
Print@(0,265) PL$
Print@(0,295) CI$
CCNT=CCNT+1:If CCNT<5 Then GoTo ENCRYPT
CCNT=0
PL$=PL$+" "
CI$=CI$+" "
GoTo ENCRYPT
ESCAPE:
Print@(0,410,2)"Press [A] to adjust rotors, [S] to change setup, [Esc] to
exit"
GoSub LETTER
'HFBmod 20150509 start - add output file
If K=18 Then 'Chr$(18+65)= "S"
```

```
    Close #1
    GoTo START
endif
If K=27 Then ' Esc key
    Close #1
end
endif
'HFBmod 20150509 end
If K<>0 Then GoTo ESCAPE
Print@(0,410)"
"
GoSub ADJR0T
'HFBmod 20150509 start - add output file
'Clear the output file after the rotors are adjusted
Close #1
gosub OpenOutFile
'HFBmod 20150509 end
GoTo CLRPAD
'HFBmod 20150509 start - add output file
OpenOutFile:
    Open OFileName$ for output as #1
    IO=0:IG=0 ' reset output character and group counts
Return
'HFBmod 20150509 end

PADFUL:
    Print@(0,410,2)"Pad Full, copy message, [Enter] to Continue  "
    GoSub LETTER
    If K<> 28 Then GoTo PADFUL
    'HFBmod 20150509 start - add output file
    print #1, Chr$(13) ' new line in output file
    IO=0:IG=0
    'HFBmod 20150509 end
    GoTo CLRPAD
REFSET:
    Print@(80,15,2)"Select the Reflector (B or C)"
    GoSub LETTER
    If K<1 Or K>2 Then GoTo REFSET
    RF=K
    Print@(80,15)Chr$(K+65);"
    Return
ROTSET:
    For J=0 To 3
        D$(J)="      ":R(J)=0:V(J)=0
    Next J
    J=9:V(0)=2:DP=30
    Print@(0,45,2)"Press [Space] to change Rotor, [Enter] for next Rotor"
ROTOR0:
    GoSub VIEW
    GoSub LETTER
    If K<>26 Then GoTo CRR0
```

```
D$(0)=RD$(J)
J=J+1:If J>10 Then J=9
GoTo ROTOR0
CRR0: If K<>28 Then GoTo ROTOR0
If D$(0)=" " Then GoTo ROTOR0
J=J-1:If J=8 Then J=10
R(0)=J:V(0)=0:V(1)=2:J=1:I=1
ROTOR:
GoSub VIEW
GoSub LETTER
If K<>26 Then GoTo CRR1
If J=R(1) Or J=R(2) Then GoTo NXT1
GoTo SKIP
NXT1: J=J+1:If J>8 Then J=1
If J=R(1) Or J=R(2) Then GoTo NXT1
SKIP: D$(I)=RD$(J)
J=J+1:If J>8 Then J=1
GoTo ROTOR
CRR1: If K<>28 Then GoTo ROTOR
If D$(I)=" " Then GoTo ROTOR
J=J-1:If J=0 Then J=8
R(I)=J:V(I)=0:V(I+1)=2:J=1
I=I+1:If I<4 Then GoTo ROTOR
Print@(0,45)"
GoSub VIEW
Return
RINGSET:
Print@(0,75,2)"Rotor Ring Setting (A to Z), [Enter] for next Rotor"
For J=0 To 3
D$(J)=" A ":RS(J)=0:V(J)=0
Next J
J=0:V(0)=2:DP=60
RING:
GoSub VIEW
GoSub LETTER
If K=28 Then GoTo CRS
If K>25 Then GoTo RING
D$(J)=" "+Chr$(K+65)+" "
RS(J)=K 'offset 0-25
GoTo RING
CRS: V(J)=0:V(J+1)=2
J=J+1:If J<4 Then GoTo RING
Print@(0,75)"
GoSub VIEW
Return
PLUGSET:
Print@(0,310,2)"PLUG ? To PLUG ? ([Enter] when all done)"
PL1: GoSub LETTER
If K=28 Then GoTo PRET
```

```
If K>25 Then GoTo PL1
Print@(30,310)Chr$(K+65)
H=K
PL2: GoSub LETTER
If K=28 Then GoTo PRET
If K>25 Then GoTo PL2
Print@(90,310)Chr$(K+65)
Pause 50
If K<>H Then GoTo SWAP
PB(PB(H))=PB(H):PB(H)=H 'plugboard letter restore
SWAP: If PB(H)<>H Or PB(K)<>K Then GoTo FORBID
PB(H)=K:PB(K)=H 'plugboard letter swap
GoSub SHOW
GoTo PLUGSET
FORBID:
Print@(0,310)"NOT ALLOWED PLUG ALREADY USED!          "
Pause 3000
GoTo PLUGSET
PRET: Print@(0,310)"                                     "
Return
SETROT:
For J=0 To 3
  D$(J)="  A ":RP(J)=0:V(J)=0
Next J
ADJR0T:
Print@(0,160,2)"Set Rotor Start Position (A-Z), [Enter] for next Rotor"
J=0:V(0)=2:DP=120
ROTP0S:
GoSub VIEW
GoSub LETTER
If K=28 Then GoTo CRP
If K>25 Then GoTo ROTP0S
RP(J)=K
D$(J)="  "+Chr$(RP(J)+65)+"  "
GoTo ROTP0S
CRP: V(J)=0:V(J+1)=2
J=J+1:If J<4 Then GoTo ROTP0S
Print@(0,160)"                                           "
GoSub VIEW
Return
VIEW:
Print@(100,DP,V(0))D$(0):Print@(150,DP,V(1))D$(1)
Print@(200,DP,V(2))D$(2):Print@(250,DP,V(3))D$(3)
DRAW:
Line(99,DP-1)-(124,DP+11),,B:Line(149,DP-1)-(174,DP+11),,B
Line(199,DP-1)-(224,DP+11),,B:Line(249,DP-1)-(274,DP+11),,B
Return
SHOW:
For J = 0 To 8
  If PB(PBD(J))<>PBD(J) Then V=2 Else V=0
  Print@((J)*36+7,330,V) "[";Chr$(PBD(J)+65);Chr$(PB(PBD(J))+65);"]"
```

```
Next J
For J = 0 To 7
  If PB(PBD(J+9))<>PBD(J+9) Then V=2 Else V=0
  Print@((J)*36+17,350,V) "[";Chr$(PBD(J+9)+65);Chr$(PB(PBD(J+9))+65);"]"
Next J
For J = 0 To 8
  If PB(PBD(J+17))<>PBD(J+17) Then V=2 Else V=0
  Print@((J)*36+2,370,V) "[";Chr$(PBD(J+17)+65);Chr$(PB(PBD(J+17))+65);"]"
Next J
Return
LETTER: 'Wait for key press
K$=Inkey$
If K$="" Then GoTo LETTER
K=Asc(K$)
If K=13 Then K=28 'CR
If K=32 Then K=26 'SP
If K>25 And K<29 Then GoTo LRET
If K>64 And K<91 Then GoTo UPC 'ucase
If K>96 And K<123 Then GoTo LOC 'lcase
GoTo LETTER
LOC: K=K-32 'conv to ucase
UPC: K=K-65 'bound 0 to 25
LRET: Return
LAMP:
For J = 0 To 8
  If PBD(J)=CODE Then V=2 Else V=0
  Print@((J)*36+10,180,V) "(";Chr$(PBD(J)+65);")"
Next J
For J = 0 To 7
  If PBD(J+9)=CODE Then V=2 Else V=0
  Print@((J)*36+20,200,V) "(";Chr$(PBD(J+9)+65);")"
Next J
For J = 0 To 8
  If PBD(J+17)=CODE Then V=2 Else V=0
  Print@((J)*36+5,220,V) "(";Chr$(PBD(J+17)+65);")"
Next J
Return
```

ENIGMAD.BAS

```
'ENIGMA for MMDOS
'written by Ray Alger May 2015
'Output file added by Hugh Buckle May 2015
OFileName$="EnigOut.txt"
Dim RD$(10):Dim D$(4):Dim R(4):Dim V(5):Dim RS(4):Dim RP(4)
Dim PB(26):Dim PBD(26):Dim REF(26):Dim KO(8,2)
Dim ROT$(10):Dim REF$(2):Dim FOR0(26):Dim REV0(26):Dim FOR1(26)
Dim REV1(26):Dim FOR2(26):Dim REV2(26):Dim FOR3(26):Dim REV3(26)
```



```

Dim S$(36):Dim T$(30):Dim BT$(4):Dim BB$(4):Dim BS$(4)
'Rotor display data
RD$(0)=" I ":RD$(1)=" II ":RD$(2)=" III":RD$(3)=" IV ":RD$(4)=" V "
RD$(5)=" VI ":RD$(6)=" VII":RD$(7)="VIII":RD$(8)="beta":RD$(9)="gamm"
'rotor knock on data
K0(0,1)=16:K0(1,1)=4:K0(2,1)=21:K0(3,1)=9:K0(4,1)=25
K0(0,2)=16:K0(1,2)=4:K0(2,2)=21:K0(3,2)=9:K0(4,2)=25
K0(5,1)=25:K0(6,1)=25:K0(7,1)=25
K0(5,2)=12:K0(6,2)=12:K0(7,2)=12
'Rotor data
ROT$(0)="EKMFLGDQVZNTOWYHXUSPAIBRCJ" 'I
ROT$(1)="AJDKSIRUXBLHWTMCQGZNPYFVOE" 'II
ROT$(2)="BDFHJLCPRTXVZNYEIWGAKMUSQO" 'III
ROT$(3)="ESOVJPZJAYQUIRHXLNFTGKDCMWB" 'IV
ROT$(4)="VZBRGITYUPSDNHLXAWMJQOFECK" 'V
ROT$(5)="JPGVOUNMFYQBENHZRDKASXLICTW" 'VI
ROT$(6)="NZJHGRCXMYSWBOUFAIVLPEKQDT" 'VII
ROT$(7)="FKQHTLXOCBJSPDZRAMENIU YGV" 'VIII
ROT$(8)="LEYJVCNIXWPBQMDRTAKZGFUHO S" 'beta
ROT$(9)="FSOKANUERHMBTIYCWLPZXV GJD" 'gamma
'Reflector data
REF$(1)="ENKQAU YWJICOPBLMDXZVFTHRGS" 'Thin B
REF$(2)="RDOB JNTKVEHMLFCWZAXGYIPSUQ" 'Thin C
'Plug board data
PB$ ="QWERTZUIOASDFGHJKPYXC VBNML"
For J=0 To 25:PBD(J)=Asc(Mid$(PB$,J+1,1))-65:Next J
DAY$="C VI I VII XEL ML YE JH WQ"
TRI$="JEX GFE"
CIPHER$="YWXUN KKCJC TLZHN SQKMD QDRKC YOIJN FJINQ KPIOU NRNG"
Function MOD26(X)
X=Cint(X) 'fix error
X=X MOD 26
If X<0 Then X=X+26
MOD26=X
End Function
Function MOD8(X)
X=X MOD 8
If X<0 Then X=X+8
MOD8=X
End Function
'DOS Screen setup
SYSTEM "mode 80,45" 'DOS box width (chars), depth (lines)
SYSTEM "title MMDOS ENIGMA" 'DOS box title
'box string data
D1$=Chr$(218):D2$=Chr$(179):D3$=Chr$(192):D4$=Chr$(196):D5$=Chr$(191):D6$=Chr$(217)
D7$=Chr$(201):D8$=Chr$(186):D9$=Chr$(200):D10$=Chr$(205):D11$=Chr$(187):D12$=Chr$(188)
BT$(0)=D1$+D4$+D4$+D4$+D4$+D5$:BT$(1)=D7$+D10$+D10$+D10$+D10$+D11$
BT$(2)=D1$+D4$+D4$+D5$:BT$(3)=" "
BB$(0)=D3$+D4$+D4$+D4$+D4$+D6$:BB$(1)=D9$+D10$+D10$+D10$+D10$+D12$

```

```
BB$(2)=D3$+D4$+D4$+D6$:BB$(3)=" "
BS$(0)=D2$:BS$(1)=D8$:BS$(2)=D2$:BS$(3)=" "
For J=1 To 47:BH$=BH$+D4$:Next J
START:
For J=1 To 36
  S$(J)=""
Next J
S$(15)=D1$+BH$+D5$
S$(25)=D3$+Left$(BH$,8)+"PLUG BOARD"+Left$(BH$,29)+D6$
S$(1)=" ***** MMDOS M4 ENIGMA *****"
T$(2)="REFLECTOR ->"
T$(4)="ROTORS USED -->"
T$(8)="RING SETTINGS ->"
T$(12)=" ROTORS -->"
For J=0 To 25:PB(J)=J:Next J 'Init Plug array
GoSub SHOW
GoSub REFSET
GoSub ROTSET
GoSub RINGSET
GoSub PLUGSET
GoSub SETROT
'HFBmod 20150509 start - add output file
  gosub OpenOutFile
'HFBmod 20150509 end

'rotor offset arrays
For J=0 To 25
  T=Asc(Mid$(ROT$(R(0)),J+1,1))-65
  U=Asc(Mid$(ROT$(R(1)),J+1,1))-65
  V=Asc(Mid$(ROT$(R(2)),J+1,1))-65
  W=Asc(Mid$(ROT$(R(3)),J+1,1))-65
  FOR0(J)=T-J
  REV0(T)=J-T
  FOR1(J)=U-J
  REV1(U)=J-U
  FOR2(J)=V-J
  REV2(V)=J-V
  FOR3(J)=W-J
  REV3(W)=J-W
Next J
'knock on data
K03A=K0(R(3),1):K03B=K0(R(3),2)
K02A=K0(R(2),1):K02B=K0(R(2),2)
'reflector data
For J=0 To 25:REF(J)=Asc(Mid$(REF$(Rf),J+1,1))-65:Next J
CLRPAD:SPL$="":SCI$=""
S$(29)="Text IN- "
S$(33)="Text OUT- "
CLPAD:S$(30)=SPL$:S$(34)=SCI$:PL$="":CI$="":CCNT=0:TB=0
```

```

S$(27)="Enter text, Press[Esc] to change settings"
GoSub SCRN
ENCRYPT:
GoSub LETTER
If K=27 Then GoTo ESCAPE
If K>25 Then GoTo ENCRYPT
TB=TB+1:If TB>200 Then GoTo PADFUL
'rotor steps
If RP(3)=K03A Or RP(3)=K03B Then GoTo ADV2
If RP(2)=K02A Or RP(2)=K02B Then GoTo ADV1 ' R2 double step
GoTo ADV3
ADV1: RP(1)=MOD26(RP(1)+1)
ADV2: RP(2)=MOD26(RP(2)+1)
ADV3: RP(3)=MOD26(RP(3)+1)
'rotor display
DP=12:D$(0)=" "+Chr$(RP(0)+65)+" ":D$(1)=" "+Chr$(RP(1)+65)+" "
D$(2)=" "+Chr$(RP(2)+65)+" ":D$(3)=" "+Chr$(RP(3)+65)+" "
'scramble
PL$=PL$+Chr$(K+65)
CODE=PB(K)
X=MOD26(CODE+RP(3)-RS(3))
X=MOD26(CODE+FOR3(X))
CODE=X
X=MOD26(X+RP(2)-RS(2))
X=MOD26(CODE+FOR2(X))
CODE=X
X=MOD26(X+RP(1)-RS(1))
X=MOD26(CODE+FOR1(X))
CODE=X
X=MOD26(X+RP(0)-RS(0))
X=MOD26(CODE+FOR0(X))
CODE=REF(X) 'Reflect
X=MOD26(CODE+RP(0)-RS(0))
X=MOD26(CODE+REV0(X))
CODE=X
X=MOD26(CODE+RP(1)-RS(1))
X=MOD26(CODE+REV1(X))
CODE=X
X=MOD26(X+RP(2)-RS(2))
X=MOD26(CODE+REV2(X))
CODE=X
X=MOD26(X+RP(3)-RS(3))
X=MOD26(CODE+REV3(X))
CODE=PB(X)
CI$=CI$+Chr$(CODE+65)
'HFBmod 20150509 start - add output file
'Write encrypted code to output file in groups of 5 characters
'New line after 12 character groups
Print #1,Chr$(CODE+65);
IO=IO+1
if IO>4 then ' Space after 5 characters

```

```

    IO=0
    IG=IG+1
    If IG>12 then
        IG=0
        Print #1, Chr$(13) ' new line
    else
        Print #1, " ";
    endif
endif
'HFBmod 20150509 end
S$(30)=PL$
S$(34)=CI$
GoSub VIEW
CCNT=CCNT+1:If CCNT<5 Then GoTo ENCRYPT
CCNT=0
PL$=PL$+" "
CI$=CI$+" "
GoTo ENCRYPT
ESCAPE:
S$(27)="Press[A] to set Rotors, [S] for setup, [Esc] to exit"
GoSub SCRN
GoSub LETTER
'HFBmod 20150509 start - add output file
If K=18 Then 'Chr$(18+65)= "S"
    Close #1
    GoTo START
endif
If K=27 Then ' Esc key
    Close #1
end
endif
'HFBmod 20150509 end
If K<>0 Then GoTo ESCAPE
S$(27)=""
GoSub ADJROT
'HFBmod 20150509 start - add output file
'Clear the output file after the rotors are adjusted
Close #1
gosub OpenOutFile
'HFBmod 20150509 end
GoTo CLRPAD
'HFBmod 20150509 start - add output file
OpenOutFile:
    Open OFileName$ for output as #1
    IO=0:IG=0 ' reset output character and group counts
Return
'HFBmod 20150509 end

PADFUL:
```

```
S$(27)="Pad Full, copy message, [Enter] to Continue"
GoSub SCRN
GoSub LETTER
If K<> 28 Then GoTo PADFUL
'HFBmod 20150509 start - add output file
print #1, Chr$(13) ' new line in output file
IO=0:IG=0
'HFBmod 20150509 end
SPL$=Right$(PL$,6)
SCI$=Right$(CI$,6)
GoTo CLPAD
REFSET:
S$(2)="REFLECTOR -> Select the Reflector [B] or [C]"
GoSub SCRN
GoSub LETTER
If K<1 Or K>2 Then GoTo REFSET
RF=K
S$(2)="REFLECTOR -> "+Chr$(K+65)
'GoSub SCRN
Return
ROTSET:
For J=0 To 3
  D$(J)="":R(J)=10:V(J)=0
Next J
J=8:V(0)=1:DP=4
S$(6)="[Space] to change Rotor, [Enter] for next Rotor"
ROTOR0:
GoSub VIEW
GoSub LETTER
If K<>26 Then GoTo CRR0
D$(0)=RD$(J)
J=J+1:If J>9 Then J=8
GoTo ROTOR0
CRR0: If K<>28 Then GoTo ROTOR0
If D$(0)=" " Then GoTo ROTOR0
J=J-1:If J=7 Then J=9
R(0)=J:V(0)=0:V(1)=1:J=0:I=1
ROTOR:
GoSub VIEW
GoSub LETTER
If K<>26 Then GoTo CRR1
If J=R(1) Or J=R(2) Then GoTo NXT1
GoTo SKIP
NXT1: J=MOD8(J+1)
If J=R(1) Or J=R(2) Then GoTo NXT1
SKIP: D$(I)=RD$(J)
J=MOD8(J+1)
GoTo ROTOR
CRR1: If K<>28 Then GoTo ROTOR
If D$(I)=" " Then GoTo ROTOR
J=MOD8(J-1)
```

```
R(I)=J:V(I)=0:V(I+1)=1:J=0
I=I+1:If I<4 Then GoTo ROTOR
S$(6)=""
GoSub VIEW
Return
RINGSET:
S$(10)="Ring Setting [A] to [Z], [Enter] for next Rotor"
For J=0 To 3
  D$(J)="  A ":RS(J)=0:V(J)=0
Next J
J=0:V(0)=1:DP=8
RING:
GoSub VIEW
GoSub LETTER
If K=28 Then GoTo CRS
If K>25 Then GoTo RING
D$(J)="  "+Chr$(K+65)+" "
RS(J)=K 'offset 0-25
GoTo RING
CRS: V(J)=0:V(J+1)=1
J=J+1:If J<4 Then GoTo RING
S$(10)=""
GoSub VIEW
Return
PLUGSET:
S$(14)="PLUG ? To PLUG ?   [Enter] when all done"
GoSub SCRN
PL1: GoSub LETTER
If K=28 Then GoTo PRET
If K>25 Then GoTo PL1
S$(14)="PLUG "+Chr$(K+65)+" To PLUG ?   [Enter] when all done"
GoSub SCRN
H=K
PL2: GoSub LETTER
If K=28 Then GoTo PRET
If K>25 Then GoTo PL2
'S$(21)=Chr$(K+65)
'Pause 50
If K<>H Then GoTo SWAP
PB(PB(H))=PB(H):PB(H)=H 'plugboard letter restore
SWAP: If PB(H)<>H Or PB(K)<>K Then GoTo FORBID
PB(H)=K:PB(K)=H 'plugboard letter swap
GoSub SHOW
GoTo PLUGSET
FORBID:
S$(14)="NOT ALLOWED PLUG ALREADY USED!"
GoSub SCRN
Pause 2000
GoTo PLUGSET
```

```

PRET: S$(14)=""
  'GoSub SCRN
  Return
SETROT:
  For J=0 To 3
    D$(J)="  A ":RP(J)=0:V(J)=0
  Next J
ADJR0T:
  S$(14)="Rotor Start Pos. [A] to [Z], [Enter] for next Rotor"
  J=0:V(0)=1:DP=12
ROTP0S:
  GoSub VIEW
  GoSub LETTER
  If K=28 Then GoTo CRP
  If K>25 Then GoTo ROTP0S
  RP(J)=K
  D$(J)="  "+Chr$(RP(J)+65)+" "
  GoTo ROTP0S
CRP: V(J)=0:V(J+1)=1
  J=J+1:If J<4 Then GoTo ROTP0S
  S$(14)=""
  GoSub VIEW
  Return
VIEW:
  DV0$=BS$(V(0))+D$(0)+BS$(V(0)):DV1$=BS$(V(1))+D$(1)+BS$(V(1))
  DV2$=BS$(V(2))+D$(2)+BS$(V(2)):DV3$=BS$(V(3))+D$(3)+BS$(V(3))
  S$(DP-1)="          "+BT$(V(0))+ "  "+BT$(V(1))+ "  "+BT$(V(2))+ "
"+BT$(V(3))
  S$(DP)=T$(DP)+"      "+DV0$+"  "+DV1$+"  "+DV2$+"  "+DV3$
  S$(DP+1)="          "+BB$(V(0))+ "  "+BB$(V(1))+ "  "+BB$(V(2))+ "
"+BB$(V(3))
  GoTo SCRN
  Return
SHOW:
  S$(16)=D2$+"  ":S$(17)=D2$+"  ":S$(18)=D2$+"  "
  For J = 0 To 8
    If PB(PBD(J))<>PBD(J) Then V=2 Else V=3
    S$(16)=S$(16)+BT$(V)+" "
    S$(17)=S$(17)+BS$(V)+Chr$(PBD(J)+65)+Chr$(PB(PBD(J))+65)+BS$(V)+" "
    S$(18)=S$(18)+BB$(V)+" "
  Next J
  S$(16)=S$(16)+D2$:S$(17)=S$(17)+D2$:S$(18)=S$(18)+D2$
  S$(19)=D2$+"  ":S$(20)=D2$+"  ":S$(21)=D2$+"  "
  For J = 9 To 16
    If PB(PBD(J))<>PBD(J) Then V=2 Else V=3
    S$(19)=S$(19)+BT$(V)+" "
    S$(20)=S$(20)+BS$(V)+Chr$(PBD(J)+65)+Chr$(PB(PBD(J))+65)+BS$(V)+" "
    S$(21)=S$(21)+BB$(V)+" "
  Next J
  S$(19)=S$(19)+"      "+D2$:S$(20)=S$(20)+"      "+D2$:S$(21)=S$(21)+"      "+D2$
  S$(22)=D2$+"  ":S$(23)=D2$+"  ":S$(24)=D2$+"  "

```

```
For J = 17 To 25
  If PB(PBD(J))<>PBD(J) Then V=2 Else V=3
  S$(22)=S$(22)+BT$(V)+" "
  S$(23)=S$(23)+BS$(V)+Chr$(PBD(J)+65)+Chr$(PB(PBD(J))+65)+BS$(V)+" "
  S$(24)=S$(24)+BB$(V)+" "
Next J
S$(22)=S$(22)+" "+D2$:S$(23)=S$(23)+" "+D2$:S$(24)=S$(24)+" "+D2$
'GoSub SCRN
Return
LETTER: 'Wait for key press
K$=Inkey$
If K$="" Then GoTo LETTER
K=Asc(K$)
If K=13 Then K=28 'CR
If K=32 Then K=26 'SP
If K>25 And K<29 Then GoTo LRET
If K>64 And K<91 Then GoTo UPC 'ucase
If K>96 And K<123 Then GoTo LOC 'lcase
GoTo LETTER
LOC: K=K-32 'conv to ucase
UPC: K=K-65 'bound 0 to 25
LRET: Return
SCRN: Cls
  For L=1 To 36
    Print S$(L)
  Next L
Return
```

ENIGMAG.bas

```
'Enigma Pseudo-random Setup Generator
'Created by Hugh Buckle May 2015
'For use with Ray Alger's ENIGMA.bas and ENIGMAD.bas
dim RD$(8),RD(8),A(26)
RD$(1)=" I ":RD$(2)=" II ":RD$(3)=" III ":RD$(4)=" IV "
RD$(5)=" V ":RD$(6)=" VI ":RD$(7)=" VII ":RD$(8)=" VIII"
Letters$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
Initialize
?
'Set the reflector to B or C
? "Reflector      = " chr$(int(rnd()*2 + 66))
SetRo R0$
? "Rotor order    = " R0$
SetRings Ring$
? "Ring setup     = " Ring$
SetPlugBoard PB$
? "Plug board     = " PB$
SetRotors RS$
```



```
? "Rotor setting = " RS$
SetMsgKey MK$
? "Message Key    = " MK$
Sub Initialize
  'Get seed and randomise
  local i,b,a$
  input "seed"; a$
  For i=1 to len(a$)
    b=b + Asc(mid$(a$,i,1))
  next
  randomize b
end sub 'Initialize
Sub SetRo(R0$)
  ' Set the wheel order
  local i,j
  i=int(rnd()*2+1)
  if i=1 then R0$="beta " else R0$="gamm "
  for i=1 to 3
    'Sets a unique rotor number (3 out of 8)
    do
      j=int(rnd()*8+1)
      loop until rd(j)=0
      rd(j)=1
      R0$=R0$+rd$(j)
    next
  end Sub 'W0
Sub SetRings(Ring$)
  'Set each of the rings
  local i,j
  for i= 1 to 4
    j=int(rnd()*26+1)
    a$=mid$(Letters$,j,1)
    Ring$=Ring$+" "+A$+" "
  next
End Sub 'SetRings
sub SetPlugBoard(Plugs$)
  'Select plugboard pairs
  local L$,NumPlugs,LettersAvail,i,j
  NumPlugs=int(rnd()*10+1)
  L$=Letters$
  LettersAvail=26
  for i=1 to NumPlugs
    GetLetter(a$,L$,LettersAvail)
    Plugs$=Plugs$+A$
    GetLetter(a$,L$,LettersAvail)
    Plugs$=Plugs$+A$+" "
  Next
End Sub 'Set PlugBoard
Sub SetRotors(RS$)
  'Create rotor initial settings
  local i,j
```

```
for i= 1 to 4
  j=int(rnd()*26+1)
  a$=mid$(Letters$,j,1)
  RS$=RS$+" "+A$+" "
next
End Sub 'SetRings
Sub SetMsgKey(MK$)
'Set a message key
For i=1 to 4
  j=int(rnd()*26+1)
  a$=mid$(Letters$,j,1)
  MK$=MK$+A$
next
MK$=MK$+" "
For i=1 to 4
  j=int(rnd()*26+1)
  a$=mid$(Letters$,j,1)
  MK$=MK$+A$
next
end Sub 'SetMsgKey
Sub GetLetter(a$,L$,LettersAvail)
'Selects a unique letter from the alphabet
local j
j=int(rnd()*LettersAvail+1)
A$=Mid$(L$,j,1)
if j>1 and j<Len(L$) then
  L$=Left$(L$,j-1)+Mid$(L$,j+1)
else
  if j=1 then
    L$=Mid$(L$,2)
  else
    L$=Left$(L$,Len(L$)-1)
  Endif
endif
LettersAvail=LettersAvail-1
End Sub 'Get letter
```

From:
<https://fruitoftheshed.com/wiki/> - FotS

Permanent link:
https://fruitoftheshed.com/wiki/doku.php?id=mmbasic_original:enigma_code_simulator

Last update: **2024/01/19 09:39**

