

HP-41 FocalMaster

Version 1.0

© 2023 Michael Göricke

Open Source

The FocalMaster is open source, you'll find it under <https://github.com/gomi42/FocalMaster> plus instruction how to use it.

Prerequisites

The FocalMaster runs in a .Net 4.8 environment which is available by default on Windows 10 and Windows.

Focal Compiler

The HP-41 Focal User Code Compiler transforms focal user code from text form to the binary raw file format or to barcodes.

Strong error checking is done during the compilation. In addition to the original HP-41 the compiler adds some features like comments and constants to structure the source file and to make it more readable and maintainable.

Source File Structure

The compiler accepts any valid HP-41 command plus parameters. Additionally the compiler recognizes the commands of some common plug-in modules and translates them to the appropriate XROM codes. Please refer to appendix A and B for more details.

The compiler is line oriented. A source code line follows the following rule set:

- 1) Strings must always be quoted with a double quote (")
- 2) Comments always start with a semicolon.
- 3) Structure of a source line
 1. A source line may be empty to structure the source code
 2. A source line may only contain a comment
 3. A source line may contain a compile directive
 4. Otherwise a source line contains a focal statement
- 4) A valid focal statement is one of the following 5 cases. Each one may be followed by a comment:
 1. A number
 2. A string
 3. An append string which in turn is a string prefixed by the single character '>'

4. A valid HP-41 command plus it's parameters
5. A valid XRom command

Example

```
define MyVariable 01
define FirstLoopLabel 2
define LoopCounter 100
define ReadyMsg "READY"
define NowMsg " NOW"

LBL "START" ;the program starts here
LoopCounter
STO MyVariable
LBL FirstLoopLabel
DSE MyVariable
GTO FirstLoopLabel
ReadyMsg
>NowMsg ;append
AVIEW
RTN
```

Directives

define

The **define** directive creates a constant, which is the association of a letter, number or text with that constant. After the constant is defined, the compiler substitutes the constant for each occurrence of the constant in the source file. **define** may occur everywhere in the source code, the constant can be used from that point of definition.

Syntax

```
define constant number
define constant letter
define constant text
```

Example

```
define MyConst 12
define MyLabel "Foo"
define MyStack Y

LBL MyLabel
```

```
RCL MyConst  
STO ind MyStack
```

XROM Codes

The compiler recognizes the commands of some common plug-in modules and translates them to the appropriate XROM codes. The list of known XROM codes can be extended by the user by adding entries to the file XRomCodes.txt. The file must be located next to the FocalComp.exe. Please refer to appendix A and B for more details.

XROM Codes

The decompiler recognizes the commands of some common plug-in modules and translates them to the appropriate XROM codes. The list of known XROM codes can be extended by the user by adding entries to the file XRomCodes.txt. The file must be located next to the FocalMaster.exe. Please refer to appendix A and B for more details.

Appendix A

Parameter

Some of the HP-41 commands require one or more parameters. The following table lists the basic parameters including their value range:

Register	integer value 0-101
Stack	single letter (case insensitive): x, y, z, t, l, m, n, o, p, q, r, a, b, c, d, e
Local_Number_Label	integer value 0-99
Local_Letter_Label	single letter (case-sensitive) without quotes: A-J, a-e
Global_Label	quoted text (e.g. "TRIANGLE")
Single_Digit	Integer value 0-9
indirect	the keyword 'ind' without quotes
Flag	integer value 0-55
XRom_Module	integer value 0-31
XRom_Function	Integer value 0-63
Comma	the character ',' without quotes

Floating Point Numbers

Floating point values may not contain blanks; the exponent follows the mantissa without blanks. The exponent is introduced by the letter 'e'.

A valid floating point value is -1234.56e-78.

The comiler also supports the special short float syntax. A float may only consist of an exponent like E56 (which is the short form of 1E56).

Parameter Type Groups

All HP-41 commands are grouped into 7 groups. The commands within each group have the same parameter structure and requirements. The following table lists the command types and their parameter options:

Type 1	no parameters required
Type 2	one the following parameter: Register Stack indirect Register indirect Stack
Type 3	one the following parameters: Single_Digit indirect Register indirect Stack
Type 4	one the following parameters: Flag indirect Register indirect Stack

Type 5	one the following parameters: Local_Number_Label Local_Label Global_Label
Type 6	one the following parameters: Local_Number_Label Local_Label Global_Label indirect Register indirect Stack
Type 7	3 parameters in the following order: XRom_Module Comma XRom_Function

Commands

Commands Type 1

NULL
 +
 -
 *
 /
 X<Y?
 X>Y?
 X<=Y?
 S+
 S-
 HMS+
 HMS-
 MOD
 %
 %CH
 P-R
 R-P
 LN
 X^2
 SQRT
 Y^X
 CHS
 E^X
 LOG
 10^X
 E^X-1
 SIN
 COS
 TAN
 ASIN
 ACOS
 ATAN
 DEC
 1/X
 ABS

FACT
X#0?
X>0?
LN1+X
X<0?
X=0?
INT
FRC
D-R
R-D
HMS
HR
RND
OCT
CLS
X<>Y
PI
CLST
R^
RDN
LASTX
CLX
X=Y?
X#Y?
SIGN
X<=0?
MEAN
SDEV
AVIEW
CLD
DEG
RAD
GRAD
ENTER
ENTER^
STOP
RTN
BEEP
CLA
ASHF
PSE
CLRG
AOFF
AON
OFF
PROMPT
ADV

Commands Type 2

RCL
STO
ST+
ST-
ST*

ST/
ISG
DSE
VIEW
SREG
ASTO
ARCL
x<>

Commands Type 3

FIX
SCI
ENG
TONE

Commands Type 4

SF
CF
FS?C
FC?C
FS?
FC?

Command Type 5

LBL

Commands Type 6

GTO
XEQ

Command Type 7

XROM

Appendix B

File format of XRomCodes.txt

The file is line oriented. Each line contains one out of the following list:

- a blank line
- a comment
- an XRom definition

A comment starts with a semicolon “;”.

An XRom definition line has the following syntax:

<function name>, <rom#>, <function#> [;<comment>]

The function name optionally can be enclosed by quotes (“”), the comment is optional. A valid XRom definition are:

```
"test1", 12, 42    ;a comment  
test2, 13, 43     ;a comment
```

Appendix C

The following functions of ROMs are known to the compiler and the compiler creates the appropriate XROM codes.

Extended I/O

-X_MASS_1A

COPYFL

DIRX

FLENG

FLTYPE

MCOPY

MCOPYPV

MVERIFY

-X_EXT_FCN

ALENGIO

ANUNDEL

ATOXL

ATOXR

ATOXX

XTOAL

XTOAR

X<>FIO

YTOAX

-X_CTRL_FNS

AID

CLRDEV

CLRLOOP

DEVL

DEVT

FINDAID

ID

INAC

INACL

INAE

INAN

INXB

INP

LOCK

NLOOP

NOTREM

OUTAC

OUTACL

OUTAE

OUTAN

OUTXB

OUTP

POLL

POLLD
POLLE
POLLUNC
RCLSEL
SRQ?
STAT
XFER
XFERC
XREFCL
XFERE
XFERN

-ADV_CTL_FN

ADROFF
ADRON
DDL
DDT
LAD
SEND
TAD
UNL
UNT

EXT-Functions
-EXT_FCN_1B

ALENG
ANUM
APPCHR
APPREC
ARCLREC
AROT
ATOX
CLFL
CLKEYS
CRFLAS
CRFLD
DELCHR
DELREC
EMDIR
FLSIZE
GETAS
GETKEY
GETP
GETR
GETREC
GETRX
GETSUB
GETX
INSCHR
INSREC
PASN
PCLPS

POSA
POSFL
PSIZE
PURFL
RCLFLAG
RCLPT
RCLPTA
REGMOVE
REGSWAP
SAVEAS
SAVEP
SAVER
SAVERX
SAVEX
SEEKPT
SEEKPTA
SIZE?
STOFLAG
X<>F
XTOA

-CX EXT-Functions

ASROOM
CLRGX
ED
EMDIRX
EMROOM
GETKEYX
RESZFL
SREG?
X=NN?
X#NN?
X<NN?
X<=NN?
X>NN?
X>=NN?

Time
-TIME 2C

ADATE
ALMCAT
ALMNOW
ATIME
ATIME24
CLK12
CLK24
CLKT
CLKTD
CLOCK
CORRECT
DATE
DATE+

DDAYS
DMY
DOW
MDY
RCLAF
RCLSW
RUNSW
SETAF
SETDATE
SETIME
SETSW
STOPSW
SW
T+X
TIME
XYZALM

-CX TIME

CLALMA
CLALMX
CLRALMS
RCLALM
SWPT

HP-IL Mass storage functions

CREATE
DIR
PURGE
READA
READK
READP
READR
READRX
READS
READSUB
RENAME
SEC
SEEKR
UNSEC
VERIFY
WRTA
WRTK
WRTP
WRTPV
WRTR
WRTRX
WRTS
ZERO

HP-IL Control Functions

AUTOIO
FINDIO

INA
IND
INSTAT
LISTEN
LOCAL
MANIO
OUTA
PWRDN
PWRUP
REMOTE
SELECT
STOPIO
TRIGGER

Wand

WNDDTA
WNDDTX
WNDLNK
WNDSUB
WNDSCN
WNDTST

-HP 41Z

$W^{1/Z}$
 W^Z
 $X^{1/Z}$
 X^Z
 $Z+$
 $Z-$
 Z^*
 $Z/$
 $Z^{1/X}$
 Z^2
 Z^3
 Z^X
 $Z=\emptyset?$
 $Z=I?$
 $Z=W?$
 $Z=WR?$
 $Z\#\emptyset?$
 $Z\#W?$
ZACOS
ZALOG
ZASIN
ZATAN
ZCOS
ZEXP
ZHACOS
ZHASIN
ZHATAN
ZHCOS
ZHSIN
ZHTAN

ZIMAG?
ZIN?
ZINT?
ZINV
ZLN
ZLOG
ZNEG
ZOUT?
ZPI*
ZREAL?
ZRND
ZSIN
ZSQRT
ZTAN
ZUNIT?
-ZSTACK
CLZ
CLZST
LASTZ
ZAVIEW
ZENTER^
Z<>
Z<>ST
ZTRP
Z<>W
ZIMAG^
ZRCL
ZRDN
ZREAL^
ZRPL^
ZRUP
ZSTO
ZVIEW
^IM/AG
1/Z
e^Z
EIZ/IZ
NXTACS
NXTASN
NXTATN
NXTLN
NXTRTN
SQRTZ
Z*I
ZCHSX
ZGEU
ZK?YN
ZKBRD
ZST+
ZST-
ZST*
ZST/
ZWLOG
-ZVECTOR
POLAR

RECT
ZARG
ZCONJ
ZMOD
ZNORM
ZPOL
ZREC
ZSIGN
ZWANG
ZWCROSS
ZWDET
ZWDIST
ZWDOT
ZWLINE

-HL ZMATH

ZAWL
ZBS#
ZCI
ZCRT
ZEI
ZERF
ZGAMMA
ZHCI
ZHGF
ZHSI
ZIBS
ZJBS
ZKBS
ZLI2
ZLIN
ZLNG
ZLRCH
ZPROOT
ZPSI
ZQRT
ZSHK1
ZSHK2
ZSI
ZSOLVE
ZWL
ZYBS
ZZETA

-SNDMTH 2x2

2^X-1
S1/N
SDGT
 SN^X
AINT
ATAN2
 $BS>D$
CBRT

CEIL
CHSYX
CROOT
CVIETA
D>BS
D>H
E3/E+
FLOOR
GMSLVR
GEU
H>D
HMS*
HMS/
LOGYX
MKEYS
P>R
QREM
QROOT
QROUT
R>P
R>S
S>R
STLINE
T>BS
VMANT
 X^3
 $X=1?$
 $X=YR?$
 $X>=0?$
 $X>=Y?$
 $Y^{1/X}$
 $Y^{^X}$
 $YX^$
-FRC
D>F
F+
F-
F*
F/
FRC?
INT?
-HYP
HACOS
HASIN
HATAN
HCOS
HSIN
HTAN
-RCL
AIRCL
 $RCL^$
 $RCL+$
 $RCL-$
 $RCL*$
 $RCL/$

-HL MATH

1/GMF
SFL
SFL\$
SFL#
BETA
CHBAP
CI
DHT
EI
ENX
ERF
FFOUR
FINTG
FLOOP
FROOT
GAMMA
HCI
HGF+
HSI
IBS
ICBT
ICGM
IERF
IGMMA
JBS
KBS
LINX
LNGM
LOBACH
PSI
PSIN
SI
SJBS
SYBS
TAYLOR
WLØ
YBS
ZETA
ZETAX
ZOUT
-PB/STs
%T
CORR
COV
CRVF
CURVE
EVEN?
GCD
LCM
LR
LRY
NCR

NPR
ODD?
PDF
PFCT
PRIME?
RAND
RGMX
RGSORT
SEEDT
ST<>S
STSORT