

MFM DISK-DRIVE EMULATOR

Get started Manual for the DE10-Nano board

Version V1.0

**Instructions for loading & flashing DE10-Nano board
running the MFM disk reader/cloner/emulator on it**

Requirement : Up and running FPGA-SoC_Linux on a
SoC/HPS board, like the DE10-Nano

Reference : DE10-Nano_User_manual.pdf

Further information on my homepage, pdp11gy.com and on
de10-nano.terasic.com/cd

It's recommended to download and install the Unix kernel

de10_nano_linux_console

Details in the manual Getting Started Guide @ de10-nano.terasic.com/cd

Jumper settings

DE10-Nano: The four slide switches (page 26, User_manual):

switch 0 : ON=Clone-Mode OFF=EMULATOR Mode

switch 1 : Type of ENCODER output: ON = mfm output,
OFF= gap recorded output (recommended).

Button 2 and 3 : Reconfigure and Reset/Restart

De0-Nano-SoC DIP switch (**SW10**) configuration, see page 12 @
User_manuel

Interface-board: 8 switches :

Switch 1: ON: LED Debug info OFF=Pattern

Switch 2 : Debug Mode ON/OFF

Switch 3-4: Unit number_

Switch 5-8 : drive typ , 16 possibilities (0 to F)

0-0-0-0 = disk_drive #0 (ST506)

0-0-0-1 = disk_drive #1 (ST412)

0-0-1-0 = disk_drive #2 (ST 225)

until: 1-1-1-1 = disk_drive #15 (= F)

Quick Start:

The firmware can be loaded in **3** different ways.

1) In the current version now works "**Load FPGA from Linux**". To load the firmware, another software is used, see

https://github.com/nhasbun/de10nano_fpga_linux_config

This software was taken over unchanged, only the Makefile was modified and the executable file is called loadrbf.

As a pure user, I recommend this method because there is no additional software required like Quartus.

Here are the steps to load the firmware and start the MFM emulator:

- Suppose you are in Folder MFM `root@socfpga:~/MFM`

- First, copy the file "soc_mfm_v1_0.zip" to the DE0-Nano-SoC board, for example, using scp or winscp. Unpack the zip file and navigate to folder soc_mfm_beta.

```
unzip soc_mfm_v1_0.zip
```

```
cd soc_mfm_v1_0
```

```
cd MFM
```

```
chmod 777 *
```

The loadrbf program is using the filename `fpga_config_file.rbf` but the RL emulator is using the file `RL_EMULATOR_SoC.rbf`. Use a link to get the issue fixed as follow:

```
ln -s ../FW/MFM_EMULATOR_SoC.rbf fpga_config_file.rbf
```

That's all !

Directory listing:

```
root@socfpga:~/soc_mfm_v1_0/MFM# ls -l
total 176
-rwxrwxrwx 1 root root  9216 Oct  5 10:15 default.dsk
-rwxrwxrwx 1 root root 43008 Oct  5 10:15 default.gap
-rwxrwxrwx 1 root root 10752 Oct  5 10:15 default.mfm
-rwxrwxrwx 1 root root   15 Oct  5 10:15 disk_speed_0.inf
-rwxrwxrwx 1 root root   15 Oct  5 10:15 disk_speed_1.inf
-rwxrwxrwx 1 root root   15 Oct  5 10:15 disk_speed_2.inf
-rwxrwxrwx 1 root root  184 Oct  5 10:15 diskinfo_0.inf
-rwxrwxrwx 1 root root  180 Oct  5 10:15 diskinfo_1.inf
-rwxrwxrwx 1 root root  169 Oct  5 10:15 diskinfo_2.inf
-rwxrwxrwx 1 root root   15 Oct  5 10:15 diskspeed.inf
lrwxrwxrwx 1 root root   26 Oct  5 10:17 fpga_config_file.rbf ->
../FW/MFM_EMULATOR_SoC.rbf
-rwxrwxrwx 1 root root 13795 Oct  5 10:15 loadrbf
-rwxrwxrwx 1 root root 32232 Oct  5 10:15 mfmemulator
-rwxrwxrwx 1 root root 31355 Oct  5 10:15 readc
root@socfpga:~/soc_mfm_v1_0/MFM#
```

Now, you can start the A) firmware loader **loadrbf** and B) the read and test program **readc** and C) the disk reader/emulator **mfmemulator**

- A) root@socfpga:~/socv2_2/RL# ./loadrbf
- B) root@socfpga:~/socv2_2/RL# ./readc
- C) root@socfpga:~/socv2_2/RL# ./mfmemulator

A) loadrbf program output:

```
root@socfpga:~/soc_mfm_v1_0/MFM# ./loadrbf
*****
MSEL Pin Config..... 0xa
FPGA State..... Powered Off
cfgwidth Register.... 0x1
cdratio Register.... 0x0
axicfgen Register... 0x0
Nconfig pull reg.... 0x0
CONF DONE..... 0x0
Ctrl.en?..... 0x0
*****
Turning FPGA Off.
Setting cdratio with 0x3.
Turning FPGA On.
Loading rbf file.
EOF reached.
*****
MSEL Pin Config..... 0xa
FPGA State..... User Phase
cfgwidth Register.... 0x1
cdratio Register.... 0x3
axicfgen Register... 0x0
Nconfig pull reg.... 0x0
CONF DONE..... 0x0
Ctrl.en?..... 0x0
*****
root@socfpga:~/soc_mfm_v1_0/MFM#
```

Now, the heartbeat LED on the interface board should be blinking

B) readc program output:

```
***** MFM-DISK read + test @ Soc/HPS *****
READ one Cylinder/Track and save it to SD card
DE10-Nano ST-506/412/225 Version V1.0
*****|*****
(c) WWW.PDP11GY.COM

>>>>> DEBUG-MODE = ON <<<<<<
>>>> Device Type = ST412 <<<<

Anzahl der Cylinder: 306
Drive_select #0 DRV_SLCTD = LOW
Drive_select #1 DRV_SLCTD = LOW
Drive_select #2 DRV_SLCTD = HIGH
READY = HIGH
SEEK_cmlpt = HIGH
TRACK_0 = LOW
DRV_SLCTD = HIGH
Drive = ready
Drive is NOT @ home
Drive positioned to home

Cylinder - nummer eingeben: 110

Trigger DataAM , (4Hex, like A5F8) :A5F8

Cylinder: 110 ,Trigger DataAM: lsb : 0xA5 msb: 0xF8

***** Step to Cylinder 110 done *****

Select Head 4

Save MFM-gaps data into file: ST412_gap-data@cylinder_110.gap
Save RAW-image data into file: ST412_raw-data@cylinder_110.dsk
Save MFM-decoded data into file: ST412_mfm-data@cylinder_110.mfm

found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 58 Nr.: 1 Gap: 58
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 629 Nr.: 2 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 1200 Nr.: 3 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 1771 Nr.: 4 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 2342 Nr.: 5 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 2913 Nr.: 6 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 3484 Nr.: 7 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 4055 Nr.: 8 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 4626 Nr.: 9 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 5197 Nr.: 10 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 5768 Nr.: 11 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 6339 Nr.: 12 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 6910 Nr.: 13 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 7481 Nr.: 14 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 8052 Nr.: 15 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 8623 Nr.: 16 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 9194 Nr.: 17 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 9765 Nr.: 18 Gap: 571
```

found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 10810 Nr.: 19 Gap: 1045
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 11381 Nr.: 20 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 11952 Nr.: 21 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 12523 Nr.: 22 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 13094 Nr.: 23 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 13665 Nr.: 24 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 14236 Nr.: 25 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 14807 Nr.: 26 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 15378 Nr.: 27 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 15949 Nr.: 28 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 16520 Nr.: 29 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 17091 Nr.: 30 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 17662 Nr.: 31 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 18233 Nr.: 32 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 18804 Nr.: 33 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 19375 Nr.: 34 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 19946 Nr.: 35 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 20517 Nr.: 36 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 21563 Nr.: 37 Gap: 1046
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 22134 Nr.: 38 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 22705 Nr.: 39 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 23276 Nr.: 40 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 23847 Nr.: 41 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 24418 Nr.: 42 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 24989 Nr.: 43 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 25560 Nr.: 44 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 26131 Nr.: 45 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 26702 Nr.: 46 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 27273 Nr.: 47 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 27844 Nr.: 48 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 28415 Nr.: 49 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 28986 Nr.: 50 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 29557 Nr.: 51 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 30128 Nr.: 52 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 30699 Nr.: 53 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 31270 Nr.: 54 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 32315 Nr.: 55 Gap: 1045
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 32886 Nr.: 56 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 33457 Nr.: 57 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 34028 Nr.: 58 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 34599 Nr.: 59 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 35170 Nr.: 60 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 35741 Nr.: 61 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 36312 Nr.: 62 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 36883 Nr.: 63 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 37454 Nr.: 64 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 38025 Nr.: 65 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 38596 Nr.: 66 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 39167 Nr.: 67 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 39738 Nr.: 68 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 40309 Nr.: 69 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 40880 Nr.: 70 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 41451 Nr.: 71 Gap: 571
found: DataAM_msb 0xA5 DataAM_lsb 0xF8 @ 42022 Nr.: 72 Gap: 571

index puls-width : 132107 = 1651.338 us
index frequency : 1344731 = 16809.137 us
Disk speed is: 3569.49 RPM

```
Select haed, 1 to 4 ( 0=exit/5=set DataAM) : 1
Decoded data      @head1 save into file: mfm-data@head=track-1_cyl-110.mfm
Recorded Gap data @head1 save into file: gap-data@head=track-1_cyl-110.gap
user raw data     @head1 save into file: raw-data@head=track-1_cyl-110.dsk
Track/head 1 : found 18 matches
```

```
Select haed, 1 to 4 ( 0=exit/5=set DataAM) : 2
Decoded data      @head2 save into file: mfm-data@head=track-2_cyl-110.mfm
Recorded Gap data @head2 save into file: gap-data@head=track-2_cyl-110.gap
user raw data     @head2 save into file: raw-data@head=track-2_cyl-110.dsk
Track/head 2 : found 18 matches
```

```
Select haed, 1 to 4 ( 0=exit/5=set DataAM) : 3
Decoded data      @head3 save into file: mfm-data@head=track-3_cyl-110.mfm
Recorded Gap data @head3 save into file: gap-data@head=track-3_cyl-110.gap
user raw data     @head3 save into file: raw-data@head=track-3_cyl-110.dsk
Track/head 3 : found 18 matches
```

```
Select haed, 1 to 4 ( 0=exit/5=set DataAM) : 4
Decoded data      @head4 save into file: mfm-data@head=track-4_cyl-110.mfm
Recorded Gap data @head4 save into file: gap-data@head=track-4_cyl-110.gap
user raw data     @head4 save into file: raw-data@head=track-4_cyl-110.dsk
Track/head 4 : found 18 matches
```

```
Select haed, 1 to 4 ( 0=exit/5=set DataAM) : 0
```

Press RESET/Button-1 for exit, Reconfig/Button-2 for restart

C) mfmemulator program output , clone mode:

```
** MFM-DISK Reader/Cloner+EMULATOR @ Soc/ HPS **
DE10-Nano ST-506/412/225 emulator Version V.1.0
*****|*****
(c) WWW.PDP11GY.COM

>>>>> DEBUG-MODE = ON <<<<<<
Disk config file: diskinfo_1.inf
# File: diskinfo_1.inf
# disk type = RD51(ST412) from DEC , 5-1/4 Zoll drive, 10Mbyte
# connected to RQDX-1 , sector size = 512 byte.

disk-data: sector-size: 512 nr. of Cylinder: 306 DataAM: A5F8
myfile3 = my_RD51(ST412)_filename.dsk
myfile4 = my_RD51(ST412)_filename.mfm
myfile5 = my_RD51(ST412)_filename.gap
*****
***** Clone-Mode *****
*****

Anzahl der Cylinder: 306
Drive_select #0 DRV_SLCTD = LOW
Drive_select #1 DRV_SLCTD = LOW
Drive_select #2 DRV_SLCTD = HIGH
READY = HIGH
SEEK_cmlpt = HIGH
TRACK_0 = LOW
DRV_SLCTD = HIGH
Drive = ready
```

Drive is NOT @ home
Drive positioned to home

index puls-width : 130106 = 1626.325 us
index frequency : 1344018 = 16800.225 us
Disk speed is: 3571.38 RPM

Cloning cylinder 305
back to home position
Save .mfm - data to SD-Card into file: my_RD51(ST412)_filename.mfm
Save .dsk - data to SD-Card into file: my_RD51(ST412)_filename.dsk
Save .gap - data to SD-Card into file: my_RD51(ST412)_filename.gap

***** Clone-Mode finished *****

Press RESET/Button-1 for exit, Reconfig/Button-2 for restart***

mfmemulator program output , emulator mode:

```
** MFM-DISK Reader/Cloner+EMULATOR @ Soc/ HPS **
DE10-Nano ST-506/412/225 emulator Version V.1.0
*****|*****
(c) WWW.PDP11GY.COM

>>>>> DEBUG-MODE = ON <<<<<<
Disk config file: diskinfo_1.inf
# File: diskinfo_1.inf
# disk type = RD51(ST412) from DEC , 5-1/4 Zoll drive, 10Mbyte
# connected to RQDX-1 , sector size = 512 byte.

disk-data: sector-size: 512 nr. of Cylinder: 306 DataAM: A5F8
myfile3 = my_RD51(ST412)_filename.dsk
myfile4 = my_RD51(ST412)_filename.mfm
myfile5 = my_RD51(ST412)_filename.gap

*****
***** Emulator-Mode *****
*****

Read MFM data file: my_RD51(ST412)_filename.mfm
Read MFM gap file: my_RD51(ST412)_filename.gap

index frequency : 1344018 = 16800.225 us
emulated disk speed is: 3571.38 RPM
gap data ENcoding

***** S T A R T ST-506/412/225 Emulator *****
Started with operating mode: 0100000110100001
```

In the Linux world you can now do smart things, like:
alias mfm='./loadrbf;sleep 2;./mfmemulator'

If you type now mfm, the firmware will be loaded and then the mfm emulator is starting.

There are **2** more ways to load the firmware to the DE10 Nano board. However, you need additional software , Quartus, Version 16.1. The DE10-Nano board is pre-configured with the Angstrom Linux - Kernel (DE10_Nano_LXDE). The default installed Linux is not able to run with a EPCS configuration.

I recommend to use the de10_nano_linux_console.img which can be very easy installed with disk-imager like win32diskimager. More details in the Getting_Started_Guide.pdf. The images and all documentation can be downloaded from www.de10-nano.terasic.com/cd .

2) Load .sof file(NOT permanent)

- De0-Nano-SoC DIP switch (SW10) to default configuration, see page 12 @ User_manual
- unzip the file "soc_mfm_beta.zip"
- Start Quartus Lite Version 16.1
- Make sure, your USB connection to the DE10-Nano is working.
- Follow the instruction in the DE10-Nano_User_manual at page 15 and load the **MFM_EMULATOR_SoC.sof** file.
- After download , the heartbeat LED schould be blinking.

3) Permanent (EPCS): Required: Quartus Lite Version 16.1

- De0-Nano-SoC DIP switch (SW10) to EPCS configuration, see page 12 @ User_manual
- unzip the file "soc_mfm_beta.zip"
- Start Quartus Lite Version 16.1
- Make sure, your USB connection to the DE10-Nano is working.
- Follow the instruction in the DE10-Nano_User_manual at page 112 and flash the DE10-Nano board with the fil **MFM_EMULATOR_SoC.jic** from folder /flash.
- After repowering the DE10-Nano board, the heartbeat LED schould be blinking.

Folders:

FW: Contains the MFM_EMULATOR_SoC.jic file for flashing the FW into the EPCS and the MFM_EMULATOR_SoC.rbf for loading the FW in the FPGA. The .cof file are configuration files if you want to convert the .sof file to .jic or .rbf by yourself.

MFM: Contains the binary runnable MFM-emulator file: **mfmemulator** and the runnable **readc** program which reads one track and/or cylinder.

The readc program is the best way to find the correct Data AM pattern.

More details in the manual **MFM_debug.pdf**

Some personal information:

I also use a Raspberry Pi 3 (model B) connected via network to the DE10-Nano board. I use the Raspberry for development purposes with a graphical interface. I can compile the programs like SIMH emulators and copy it to the DE10-Nano board, because it is binary compatible. That's so great and there is still a lot of room for further additional applications.

Instructions: Rebuild the MFM-emulator running on DE10-Nano board.

Firmware:

Use Quartus V16.1 and open the Project RL_emulator.qpf

After compiling the Project, use the the MAKE_jic.cof and MAKE_rbf.cof file to build the .jic and .rbf files.

Programming environment:

It was difficult to make everything runnable because many things in the documentation and in the examples were not correct. Here is a step by step explanation to rebuild the MFM-emulator if necessary or if you want to design some add-on application.

- Download and install **Quartus Version 16.1.**
- Download and install Intel **SoCEDSPRO Version 16.1**

Fix Problems:

*1 : error You must define soc_cv_av or soc_a10 before compiling with HwLibs
Go to intelFPGA/16.1/embedded/ip/altera/hps/altera_hps/hwlib/include
Copy all .h files in the folder soc_cv_av and soc_a10

*2 : generate_hps_qsys_header.sh : PATH is not set correct: correct as following:
#!/bin/sh
PATH=/cygdrive/C/altera_lite/16.1/quartus/sopc_builder/bin:\$PATH
sopc-create-header-files \
"\$PWD/RL_system.sopcinfo" \
--single hps_0.h \
--module hps_0

*3: Modify the makefiles, here the MFM-emulator cylinder-read make file
software/MFM/Makefile // mfmemulator
software/read/Makefile // readc

mfmemulator makefile:

```
#
TARGET = mfmemulator
ALT_DEVICE_FAMILY ?= soc_cv_av
ALT_DEVICE_FAMILY ?= soc_a10
#
CROSS_COMPILE = arm-linux-gnueabi-
#CFLAGS = -static -g -Wall -I$
{SOCEDS_DEST_ROOT}/ip/altera/hps/altera_hps/hwlib/include
CFLAGS = -g -Wall -I$
{SOCEDS_DEST_ROOT}/ip/altera/hps/altera_hps/hwlib/include/$
{ALT_DEVICE_FAMILY} -Dsoc_cv_av -Dsoc_a10
LDFLAGS = -g -Wall
CC = $(CROSS_COMPILE)gcc
ARCH= arm

build: $(TARGET)
$(TARGET): main.o
    $(CC) $(LDFLAGS) $^ -o $@
%.o : %.c
    $(CC) $(CFLAGS) -c $< -o $@

.PHONY: clean
clean:
    rm -f $(TARGET) *.a *.o *~
```

References:

User Manual : [MFM-disk-emulator.pdf](#)

<http://www.pdp11gy.com>

<https://github.com/pdp11gy/SoC-HPS-based-MFM-disk-emulator>

<https://github.com/pdp11gy/SoC-HPS-based-RL-disk-emulator>

<http://www.pdp11gy.com/sddoneE.html>

Sources @ GitHub:

All sources are included in the Folder **MFM-disk_Emulator_SoC_v1_0.zip**

It's based on the same Qartus V16.1 environment.

For comments and questions, please contact me.

INFO@pdp11gy.com