DEC RL01/RL02 DISK-DRIVE EMULATOR

User Manual for the DE10-Nano board

Version 2.2

DE10-Nano board with emulator interface
SoC/HPS environment: Cyclone V FPGA + ARM Cortex-A9 CPU.

Emulates up to 4 RL01/RL02 drives simultaneously
Supports mixed environment of emulated + real RL drives
Access to 16 x 4 RL01/RL02 configurations sets
Support .DSK data format
Open FPGA-SoC-Linux environment
SoC/HPS based disk emulator for the DEC RL01 and RL02 disk drives

CONTENTS

Secure the vintage software and preserve it on new technology

Overview:

Project Start was in 2009. In the initial phase, a PIC processor and shortly afterwards an ARM SBC was used. This idea quickly turned out to be unrealizable and I walked step by step into the FPGA world. First, I had worked on the MAX-2 CPLD. The realization failed due to the non-existent onchip-memory. Then the DE1 board was used with the CYCLON 2 FPGA. The RL01 emulator in the first version was completed in **2012** (see video). Then, the next versions were realized with the DE0-Nano, BeMicro CV board and it was now possible to emulate up to 4 RL02 disk drives simultaneously. Unfortunately, the BeMicro CV board is no longer available until now (JAN 2017), to bad and it was a big setback. The current version was ported to the DE10-Lite board and **many** new options have been developed, such as basics of WLAN support. The technical development continues. The DE10-Lite board is too slow due to the NIOSII CPU and the licensing policy of (ex)Altera. The faster NIOSII CPU is licensed. Also, the SD card is too slow because it works in SPI mode and another software driver would cause license costs again. In addition, the Quartus Lite version higher than 16.1 is also limited. (SD-RAM). The entire application has now been ported to a **SoC / HPS** environment (July 2018).

NEW

The complete application is now ported to an open Linux SoC/HPS environment. In my case, it is the DE10 Nano board with Cyclon-V FPGA and 800MHz dual-core ARM Cortex-A9 processor. The performance increase is impressive. For example: Reading 4 RL images only takes about 5 seconds. With this project, much more is possible in addition to the RL-emulator. It is now possible to run my RL-emulator and all SIMH emulators on a single board. A PDP-11, PDP-8 and VAX emulator with all available RL-based software is running on this “one hand” large board. A Raspberry Pi 3-B connected via network can be used for development purposes with the graphical interface. For example: You can compile the programs like SIMH CPU-emulators and copy it to the DE10-Nano board, because it is binary compatible!

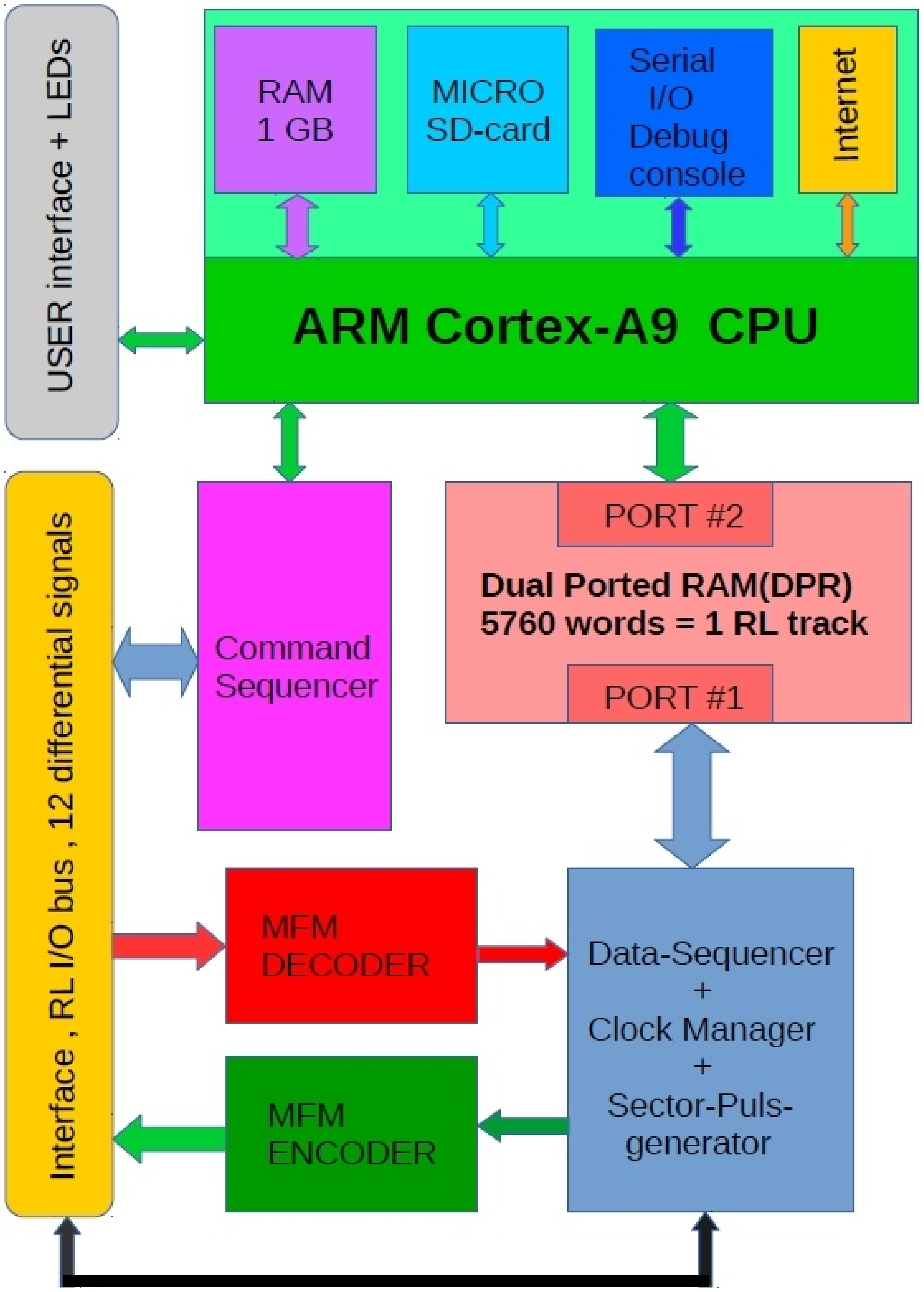
Architecture:

Basically, the design of my DEC RL02/RL01 disk drive emulator works like a Solid-State-Disk(SSD), interfacing the DEC RL-disk serial bus signals (1980) to the current FPGA technology. The heart of my design is a DPR (Dual Ported RAM) which can hold one RL-track. DPR-Port #1 is responsible for the firmware communication like MFM De/En-coding, provides the complete data transfer to/from DPR-Port #1 based on a data sequencer and runs completely automatically. DPR-Port #2 is responsible for the data transfer to/from the (NIOSII) CPU. Sounds easy, but it was very difficult to construct the right data format emulating the cartridge format with CRC and all the servo information. The (NIOSII) CPU is also responsible for the data transfer in the memory with up to 4 emulated RL drives and finally also for the transfer to/from the SD card. The operation of the RL02/RL01 emulator is best viewed with a VIDEO via YouTube, however in the first version from **2012**, based on the DE1-Board.

https://www.youtube.com/watch?v=0i3ypBU39as

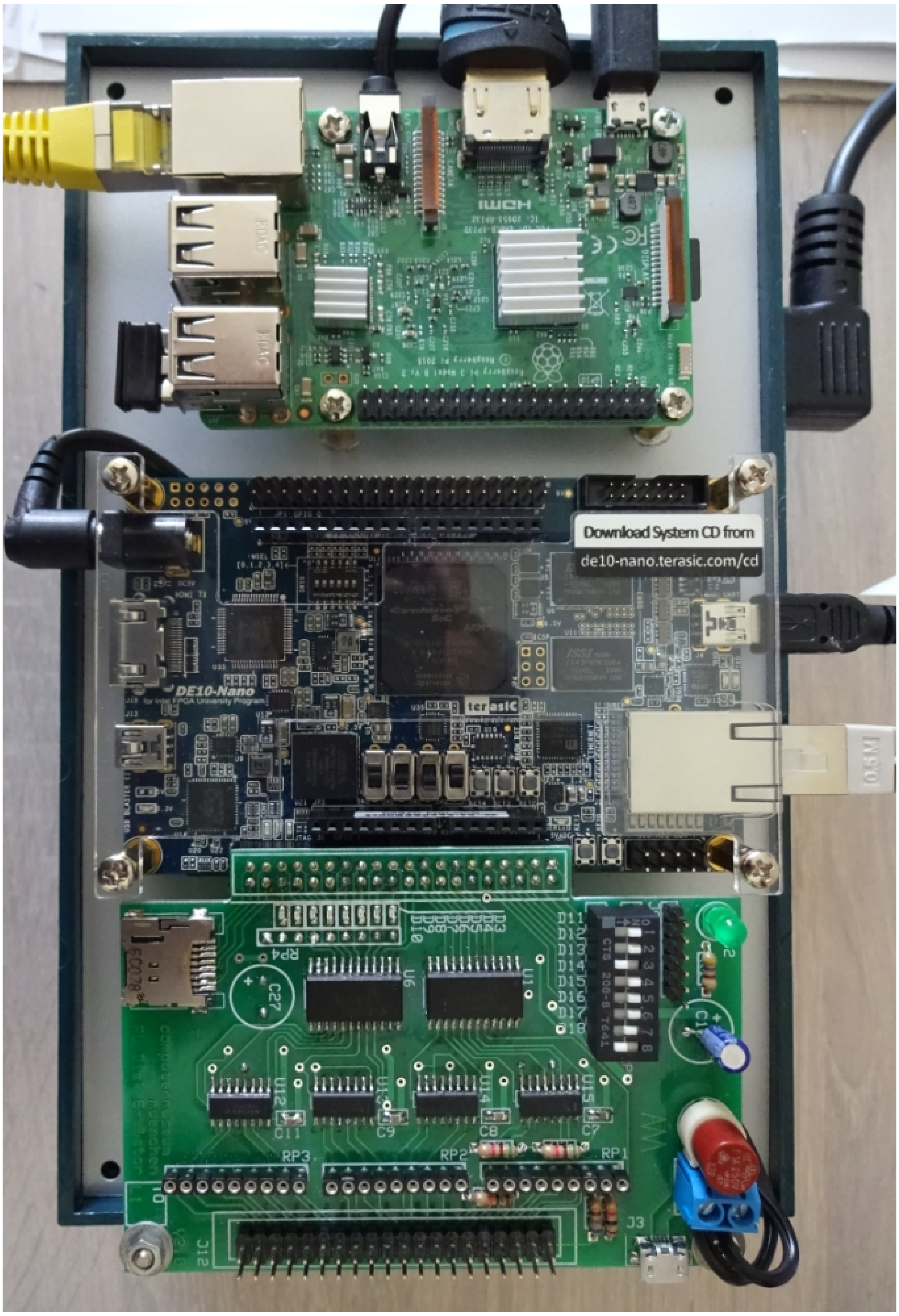
Next page: Block diagram architecture overview

Architecture: DE10-Nano based RL Emulator


USER interface + LEDs
RAM 1 GB
MICRO SD-card
Serial I/O Debug console
Internet
ARM Cortex-A9 CPU
Interface , RL I/O bus , 12 differential signals
Command Sequencer
PORT #2
Dual Ported RAM(DPR) 5760 words = 1 RL track
PORT #1
MFM DECODER
MFM ENCODER
Data-Sequencer + Clock Manager + Sector-Puls-generator

WWW.PDP11GY.COM

Emulated Data Center from 1980



These 3 components are needed to emulate a data center from the years around 1980:

Raspberry Pi 3-B , DE10-Nano board and RL-emulator interface board

Furthermore, there is still plenty of room for other applications, such as an MFM disk emulator. The second PIO and the Arduino connector are still unused.

I would be glad about a collaboration

Data format

The DEC RL01/RL02 disk drive did have a capacity of 5.2MB/10.4MB, 2 Heads(surfaces), 256/512 cylinder, 40 sectors/track. 1 sector contains 128 16-bit words (256 Byte) of Data + 12 16-Bit words for Servo/Header/CRC Data = 140 words(280 Byte)/sector. The emulater is using the .DEC format which contains all the information plus a serial number and the bad sector file. Another disk format is the disk image structure **.DSK** which is used for CPU emulators.

This format is full supported now and implemented inline. At write operation, the .DEC file and the .DSK file will be written. At read operation, first try is to read the .DEC file. If it does not exist, the .DSK file will be read.

RL02: .DSK file = 10485760 byte .DEC file = 11796992 byte
RL01: .DSK file = 5242880 byte .DEC file = 5898752 byte

The .DEC files are **compatible** to all my other RL-emulators.
Best interface to the SIMH project.

Version 2.x Release notes

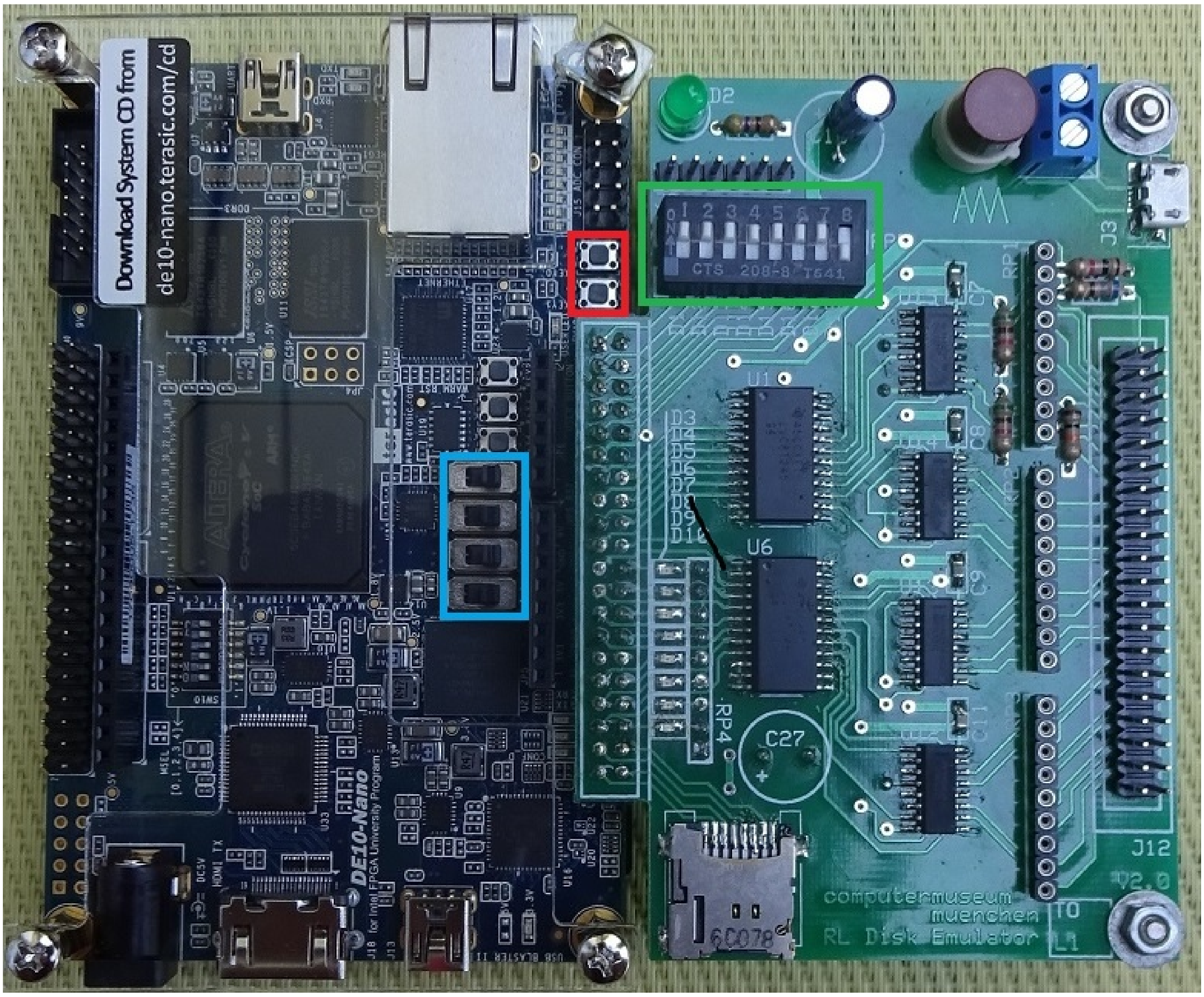
Features & Enhancements

Emulated cartridge SERIAL NUMBER (SN) handling:
Up to the version 1.5, the handling of the cartridge serial numbers was static, and by default, always the same serial number was used. This can result in errors by some DEC operating systems. In version 2.0, the cartridge serial number can be set with the content of the file **SN**x**.TXT** and can be changed individually for each subset at any time with a text editor. It contains the serial numbers for each 4 cartridges per disk-subset (DL0: to DL3:) in the form of 2 16-bit integer values (HEX-notation). As long as the file **SN**x**.TXT** is present, the serial number with the values from the file **SN**x**.TXT** will be always set after loading the RL images. If this is no longer necessary, then simply delete the file **SN**x**.TXT.** Now, the serial numbers of disk image are used. The cartridge serial number is located on the last cylinder, RL01=256, RL02=512. You can also check the serial number with a HEX editor by opening a RL02 emulator image file and navigating to the offset (h) B3A610. For example, if serial number is 1234 and 5678
00B3A610 00 00 00 80 34 12 78 56 00 00 00 00 FF FF FF FF
Please do not use a (hex) editor to change the SN. It would not build the new data CRC and would therefore cause problems, like boot/dup error.

Customize the disk-subset environment:
To make the operation and documentation easier, a configuration file **RL**x**.TXT** for each disk subset is now implemented. In this file, you can write your own notes with a standard text editor.

Configuration & Jumper-Settings

DE10-Nano http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=1046



slide switches 0-3 : select one of 16 disk set: 0 to F

Button 1 Reset / Restart after Reset
Button 2 Reconfigure / Exit after Reset

SW-0 (Nr.8) - SW-7(Nr.1) :

SW-0(Nr.8)	**Initialize** a new disk subset , selected by the slide switches
SW-1(Nr.7)	**Force power OK**
SW-2(Nr.6)	**Debug mode ON/OFF**
SW-3(Nr.5)	RL drive type, **RL01 or RL02** (ON)

SW-7 – SW-4 : 4 disk units, **DL3: - DL0:** will be selected and configured.
All 4 switches OFF = OFFLINE mode.

Implementation/architecture of the Interface board:

The interface board consists of the following components:

2	LVC8T245	= level converter
2	AM26LS31	= Transmitter
2	AM26LS32	= Receiver

8 LEDs
1 8 pin DIP-switch
3 pluggable resistor networks
1 holder for a micro SD card
2 connectors (40 pin)
1 6 pin connector for serial connection with 19200 baud based on + 3.3 Volt.
A "RoHS TTL-232R-3V3" USB converter will provide PC-connection.
2 5 Volt Power distributen connectors micro-USB connector: This is a simple and inexpensive way for a battery backup implementation with a standard handy-power bank.

Interface LED's (from right to left):

LED 0	heartbeat (blinking)
LED 1	Power OK
LED 2	Read/Seek in progress
LED 3	Write in progress
LED 4	Configured Unit dl3 active
LED 5	Configured Unit dl2 active
LED 6	Configured Unit dl1 active
LED 7	Configured Unit dl0 active

Pluggable resistor networks:

Necessary if the interface board is connected directly to the RL controller.

Serial Interface:

The serial interface is configured for **19200** baud based on a 6 pin connector with + 3.3 Volt.
A "RoHS TTL-232R-3V3" USB converter will provide PC-connection.

Battery Backup:

The additional micro-USB connector is available for connecting a standard Handy Power Bank. This is a very simple and cost-effective Battery Backup implementation.

Micro-SD

Is not used and is replaced by the onboard Micro-SD from the DE10-Nano board.

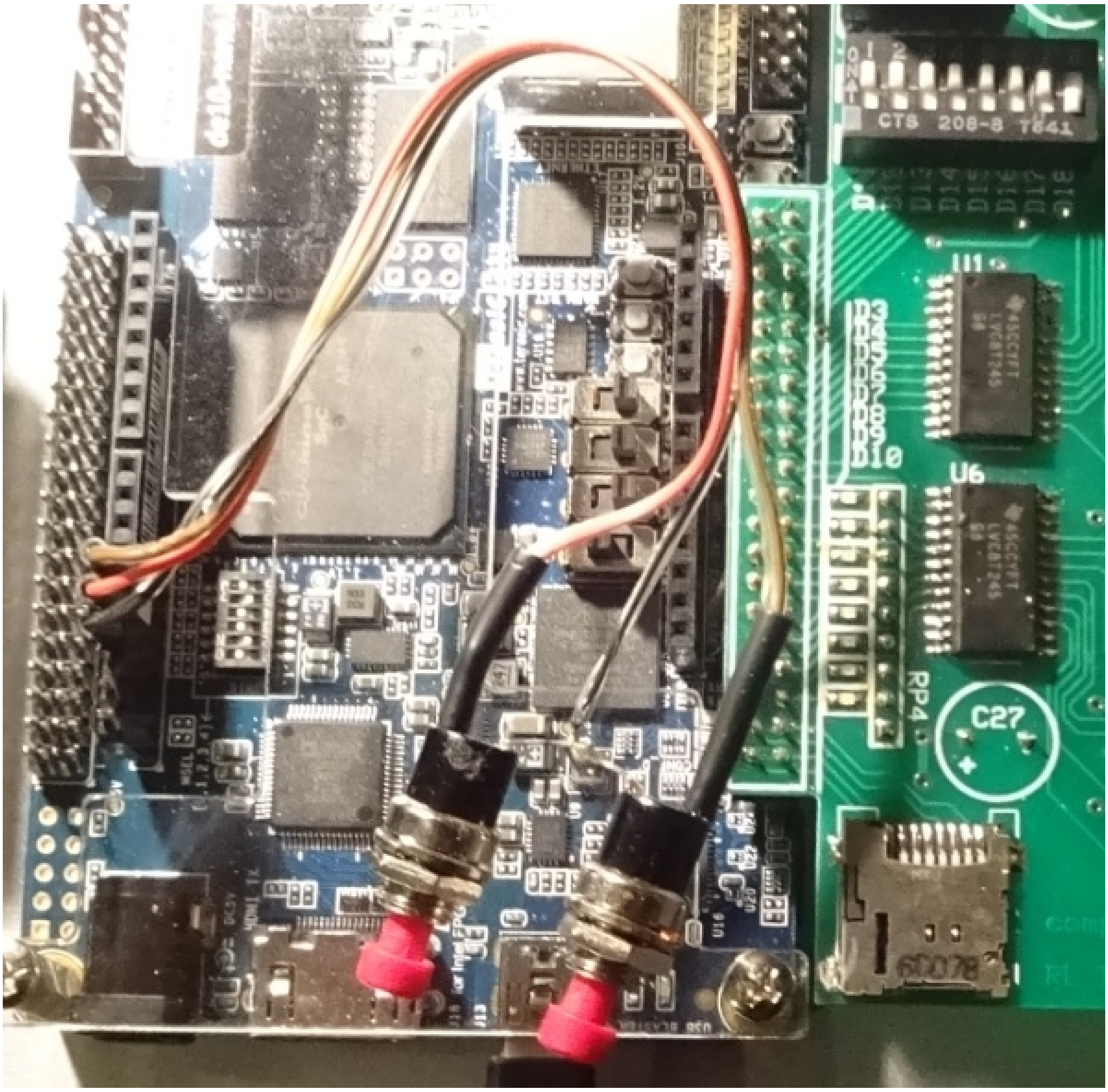
A **RL02** emulator-image file has a size of **11796992 KB** with file extension "**.DEC**"
A **RL01** emulator-image file has a size of **5898752 KB** with file extension "**.DEC**"

2.2 Reset/Reconfig buttons

Unfortunately, the reset and reconfig buttons 1 and 2 on the DE10 Nano board are very small and difficult to reach. Now it is possible to control the reset/reconfig function alternatively via 2 external button. These buttons must be connected to the Arduino connector as follows:

Arduino_IO13 = AH12 (Button 1) = reset/exit
Arduino_IO12 = AH11 (Button 2) = reconfig/restart
See also DE10 User Guide 3.6.3 Arduino Uno R3 Expansion Header , page 30

Design example:

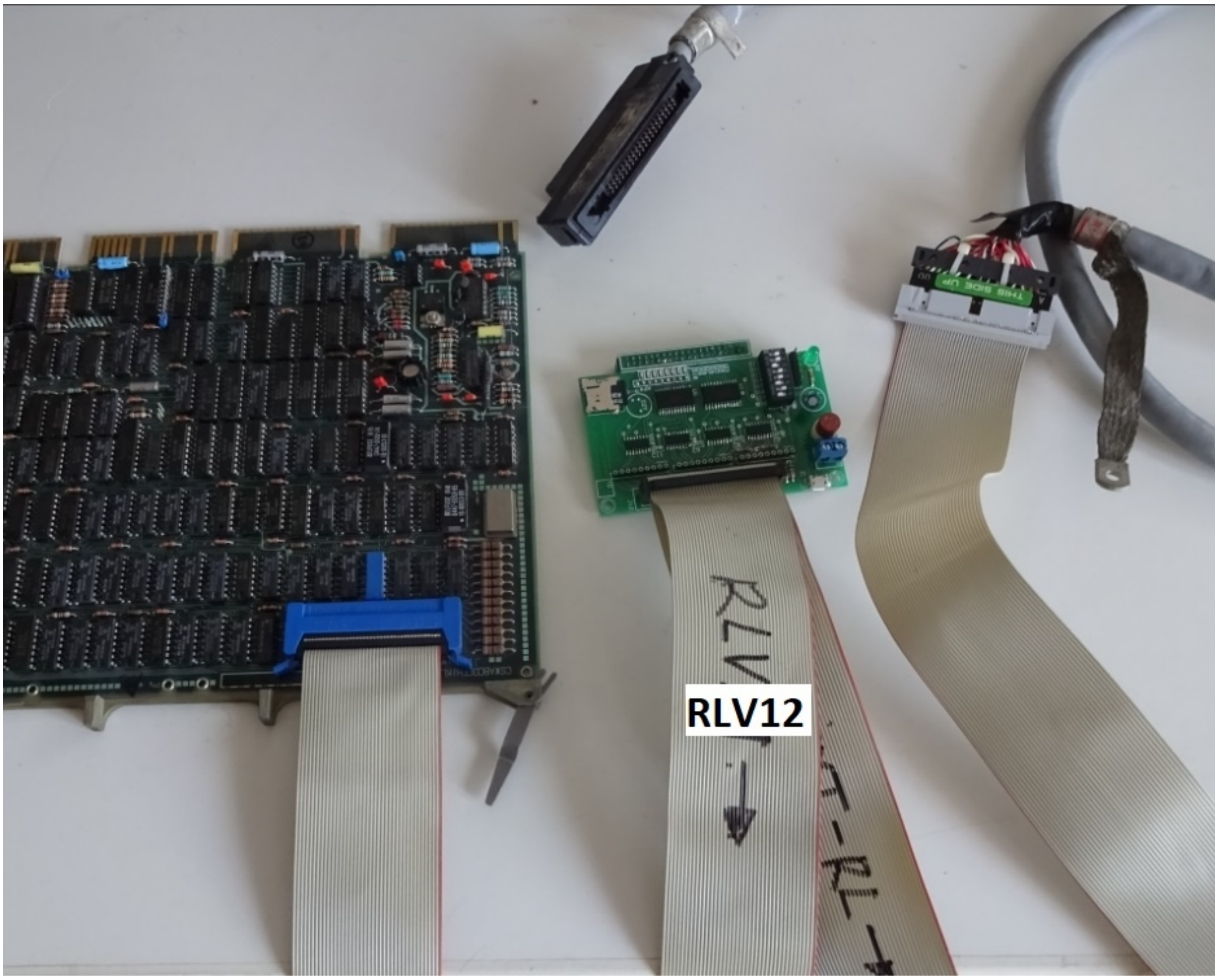


3. Environment and Startup

Overview of the hardware and software setup including step-by-step procedures from installing the necessary software tools to use the DE10-Lite board.

This example shows a Q-BUS implementation with RLV12 controller

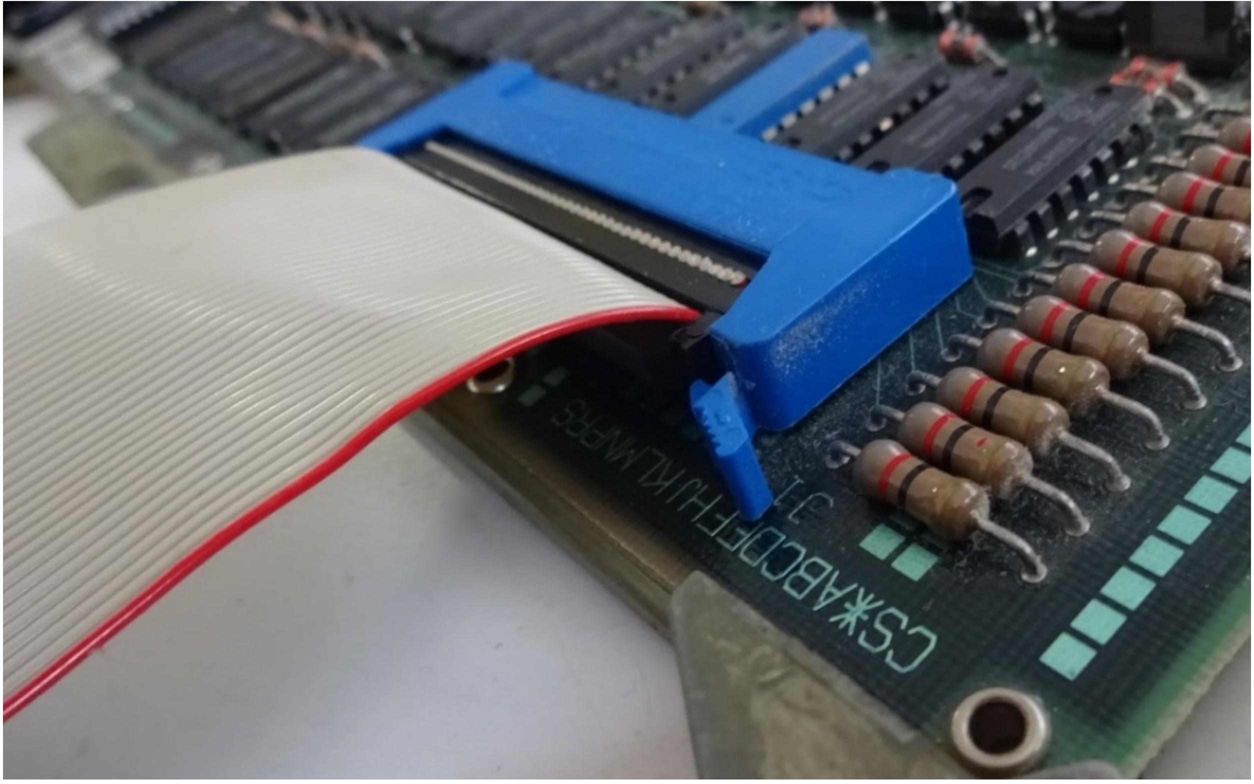
The following figure shows the connections based on a RLV12 Q-BUS controller-board to the emulator board and to an external RL disk drive.



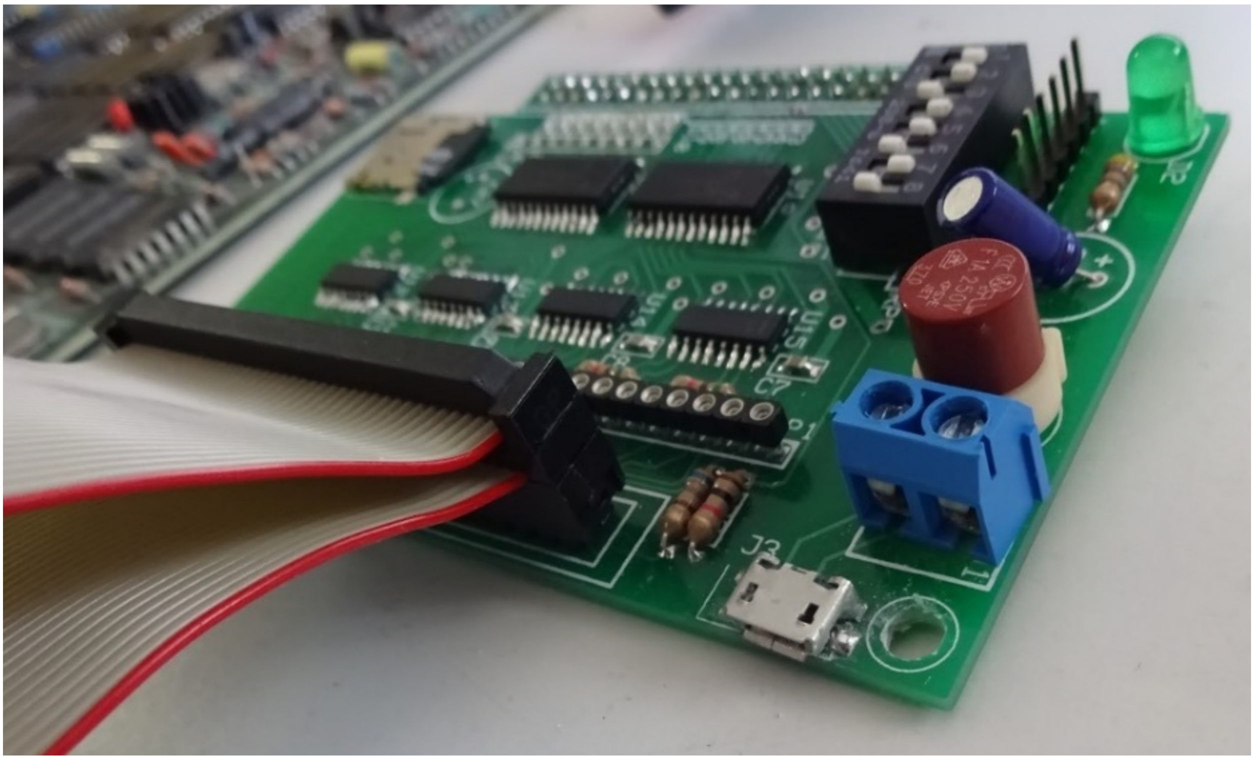
A 40-pin flat ribbon cable is required for the connections

The details of the connections are shown in the next 3 pictures

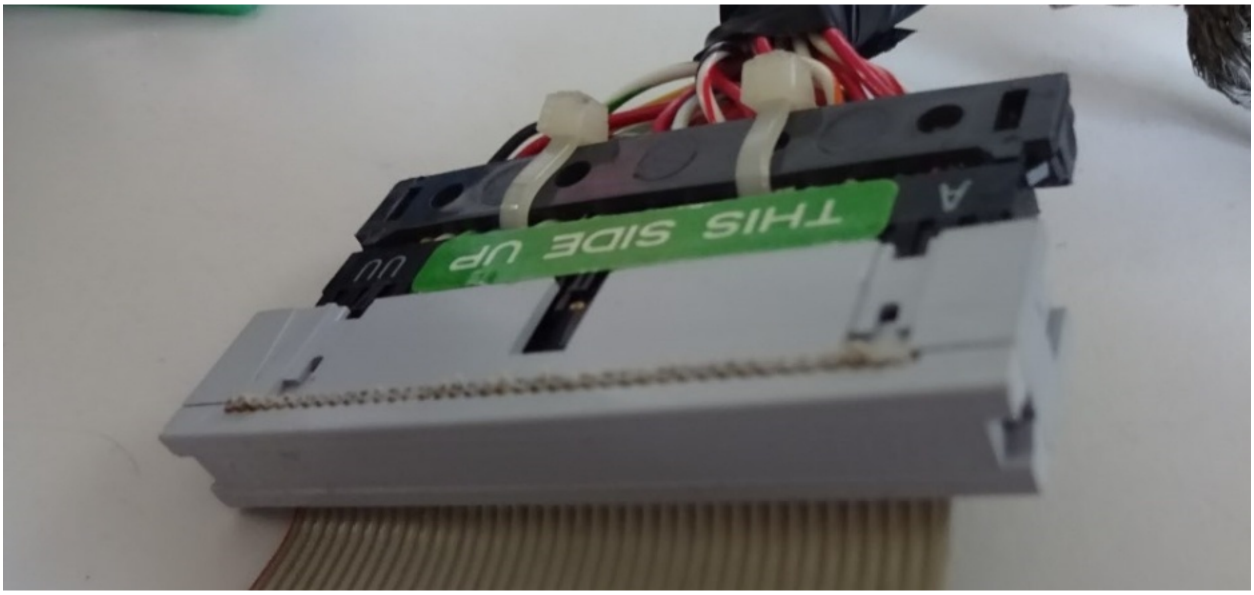
RLV12 (RLV11, RL11, RL8) **connection:**



Emulator board connection:



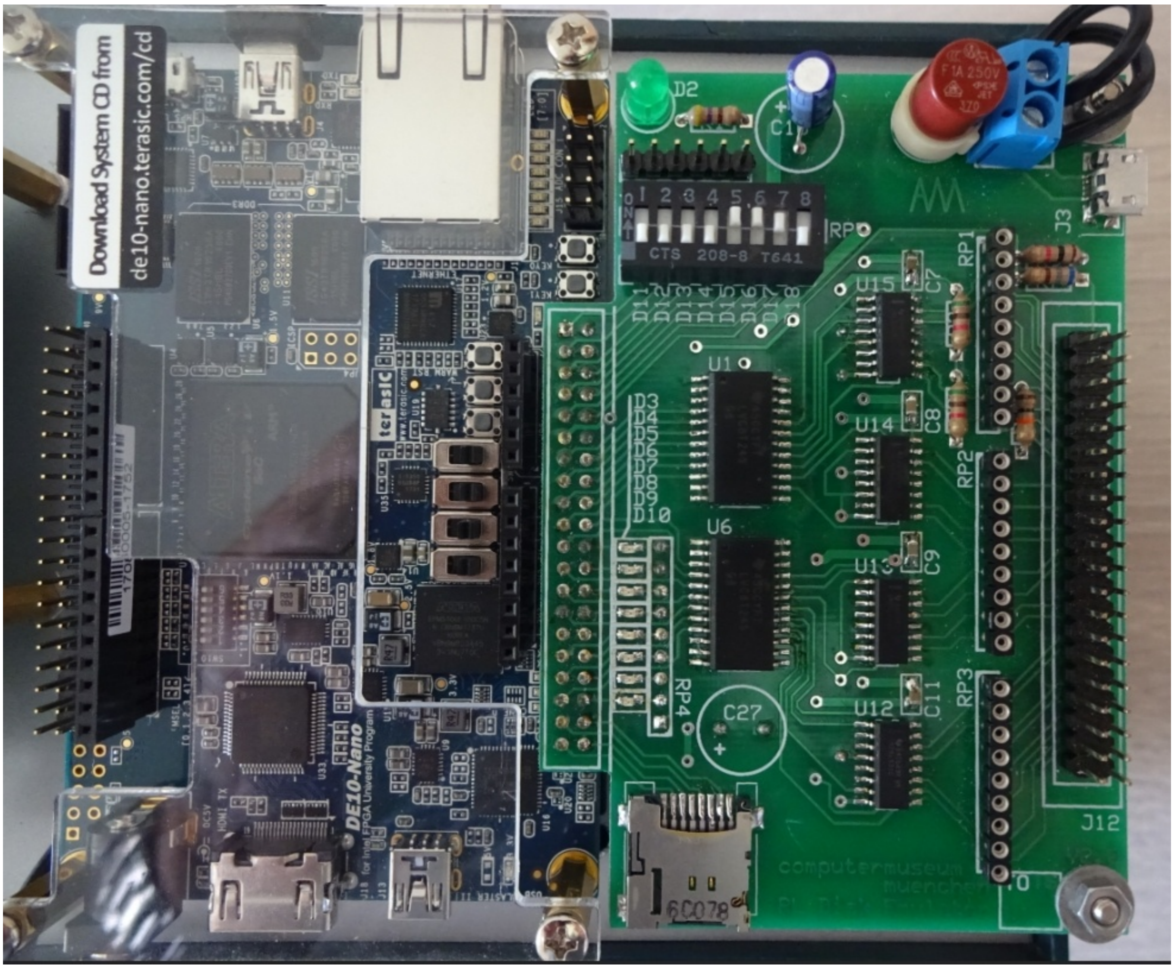
RL-BUS connection:



Disconnect if no external RL-drive is used , but install the 3 terminator resistor-networks on the emulator board.

Jumper settings for start-up/test example, NO external connection is required.

OFFLINE mode, DEBUG mode, drive-type=RL02, configured device: none, Force POK



Jumper Settings details on page 7

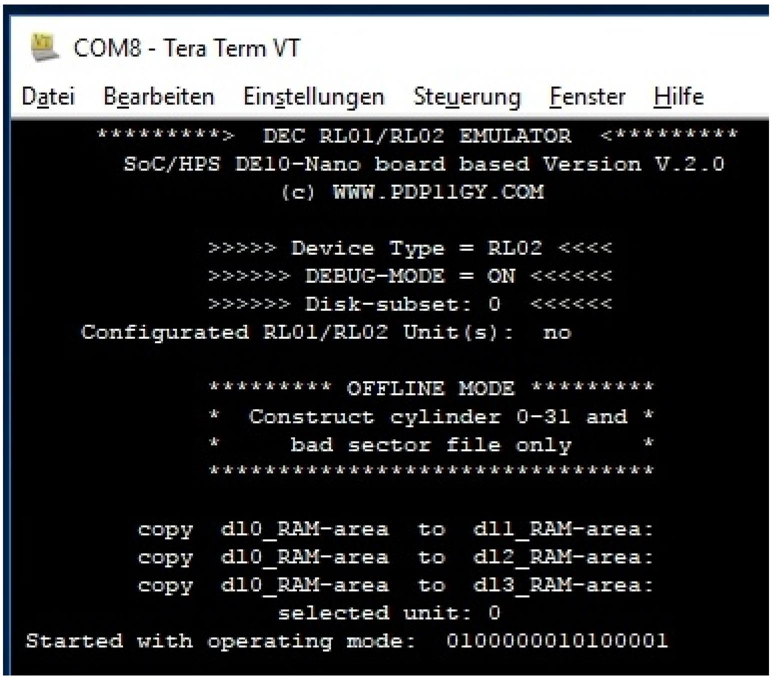
<u>Steps to bring up the emulator interface board up and running</u>

Prerequisite

Download file **socv2.zip**(or higher**)** from my homepage, http://www.pdp11gy.com/doneE.html or from GitHub: https://github.com/pdp11gy/SoC-HPS-based-RL-disk-emulator . Unzip this file and follow the instructions in the README.txt file.
Download the manuals and all reference files from: de10-nano.terasic.com/cd

The RL emulator software was developed with Quartus version V16.1. The project is flexible designed based on 2 areas, the FirmWare(FW) part and the runable part, the actual program. When the FW has successfully started, the heartbeat LED on the interface board is flashing. There are at least 3 different methods available for loading the FW. Loading/flash the .sof or .jic file located in socv2/FW requires the Programmer-tool from Quartus Lite or a portable Programmer Version. The 3rd method, loading the .rbf file is pure Linux stuff: FPGA configured from HPS software. Please follow the instructions of terasic. At this point in time, I would recommend flashing the RL_EMULATOR_SoC.jic. file with configuration: FPGA configured from EPS (1-0-1-1-0-N/A) using the Linux Console (kernel 4.5) Version 1.3 .If a "easy of use" method for the 3rd possibility is available, I will publish it. Now, copy the file sov2.zip to the DE10-Nano board using scp for example (scp socv2.zip root@xxxx.xxxx.xxxx.xxx:/home/root). Unzip the file and navigate to folder socv2/RL. Change mode via chmod 777 * and start the emulator with the command: ./rlemulator

The system is ready for use and should start as follows:



```
      *********>  DEC RL01/RL02 EMULATOR  <*********
        SoC/HPS DE10-Nano board based Version V.2.0
                    (c) WWW.PDP11GY.COM

               >>>>> Device Type = RL02 <<<<
               >>>>>> DEBUG-MODE = ON <<<<<<
               >>>>>> Disk-subset: 0  <<<<<<
     Configurated RL01/RL02 Unit(s):  no

               ********* OFFLINE MODE *********
               *  Construct cylinder 0-31 and *
               *     bad sector file only     *
               ********************************

         copy  dl0_RAM-area  to  dl1_RAM-area:
         copy  dl0_RAM-area  to  dl2_RAM-area:
         copy  dl0_RAM-area  to  dl3_RAM-area:
                    selected unit: 0
 Started with operating mode:  010000001010001
```

Power up :

The heartbeat LED is blinking.

It takes about 10 seconds to start the Linux system. After starting the rlemulator, the 8 LED's show a quick back and forth run which means the rlemulator has been started and the communication between FPGA and HPS is working fine.

As shown in the last picture, the system is now configured
Depending on Online or **Offline** mode, a different LED pattern is started

Offline Mode:

In this operating mode, no complete RL drives are emulated, access to the SD card is not possible and the emulator can be started without external connections, primary for verify purpose. **BUT,** if you connect the RL-Bus to the emulator board :
Access to an external "real" RL drives is possible (for test/verify purpose the external cable)
Limited access to cylinder 0-31 only is also possible. (about 0.3 MB)

Assuming RT-11 runs from another drive, such as RX01, RX02 or RX50, alternatively, my bootable RT-11 image files are available from my homepage, then the following commands can be used without problems (in this hardware example) :

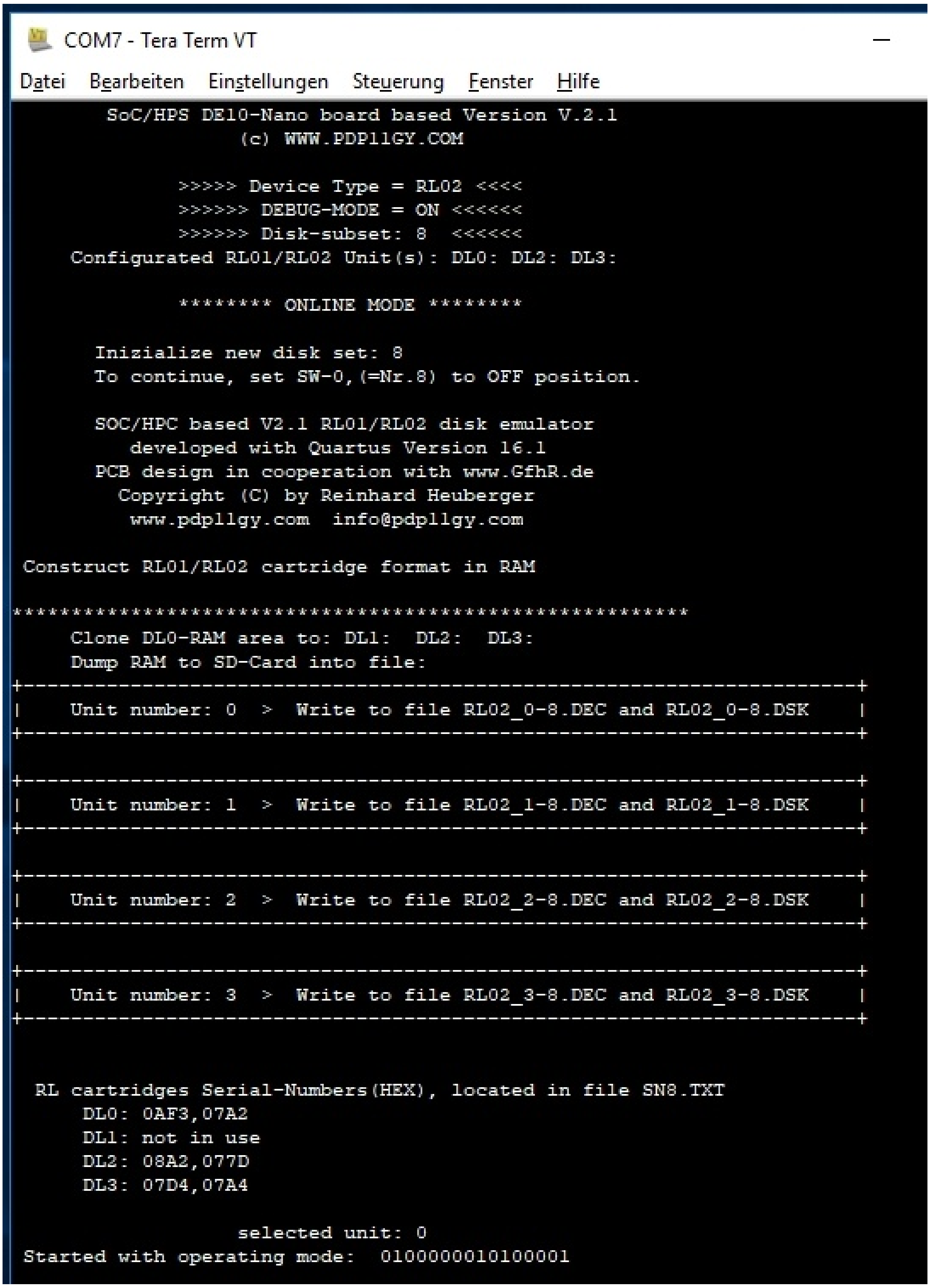
dump/term dl0:	(or dl2: , dl3:)	
dump/term/only:23730		**// get the cartridge SN**
init dl0:	(or dl2: , dl3:)	
copy/sys *.* dl0:	(or dl2: , dl3:)	(**cancellation after 0.3 MB**)
dir dl1:	(external , "real" RL02)	

Online Mode:

At leased one of the 4 SWITCHES SW-4 -SW-7 is ON: **Online mode is selected**

4. SELECT + INIT mode

With the implementation of the Select mode, 16 disk sets, each consisting of a maximum of 4 RL-images are supported. This results in a total of 16 sets and means that a maximum of 64 RL-images are available and accessable in sets of 4 RL-images. Of course you can extend this as you like because it is a Linux environment.
The SELECT + INIT mode is activated with SW-0 on the interface board. Please note: 4 files are always created for DL0: to DL3:
Assuming the **slide switches** are set to ON-OFF-OFF-OFF , disk set **8** will be used as in the following picture:

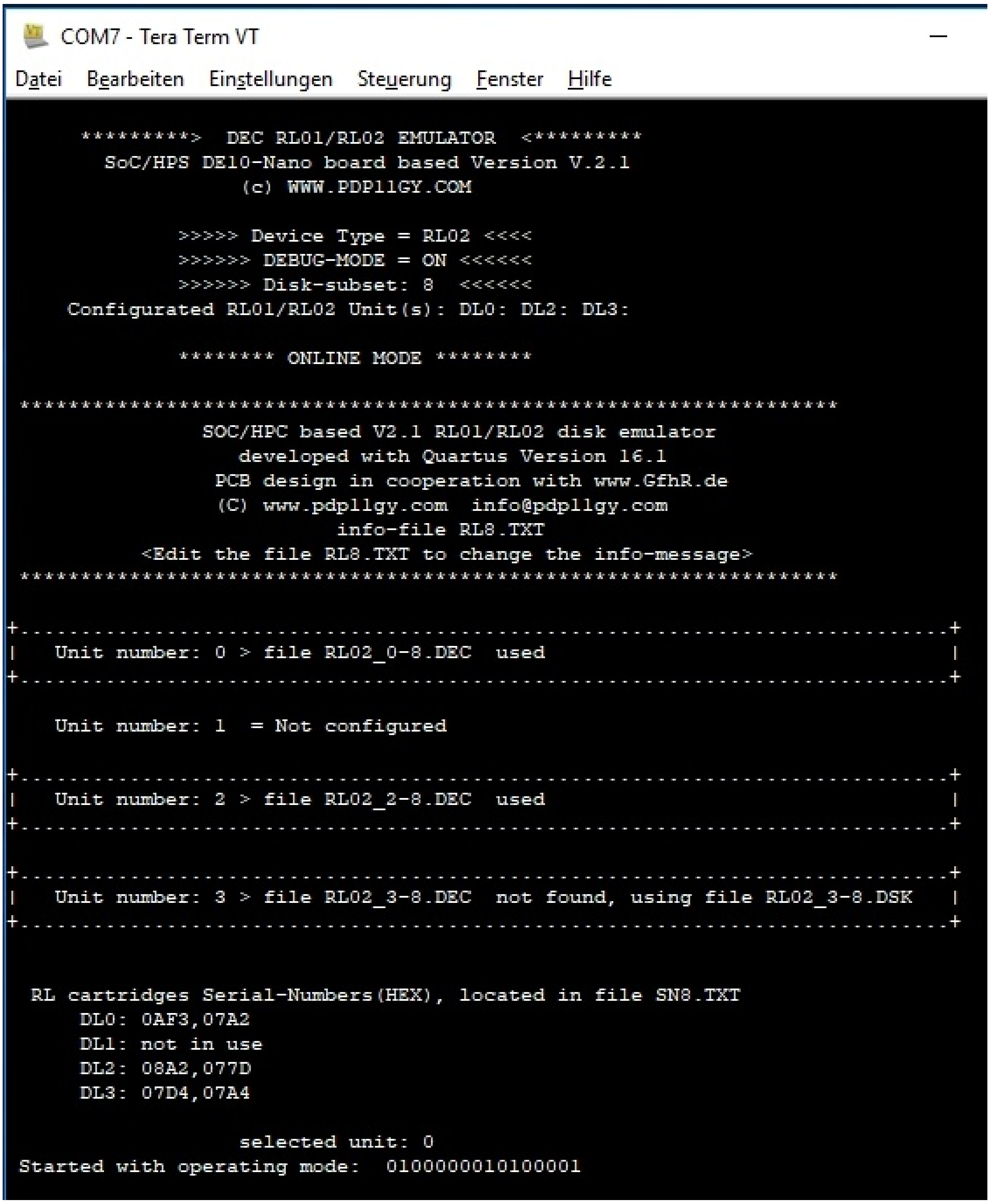


```
        SoC/HPS DE10-Nano board based Version V.2.1
                   (c) WWW.PDP11GY.COM

              >>>>> Device Type = RL02 <<<<
              >>>>>> DEBUG-MODE = ON <<<<<<
              >>>>>> Disk-subset: 8  <<<<<<
     Configurated RL01/RL02 Unit(s): DL0: DL2: DL3:

              ******** ONLINE MODE ********

       Inizialize new disk set: 8
       To continue, set SW-0,(=Nr.8) to OFF position.

       SOC/HPC based V2.1 RL01/RL02 disk emulator
          developed with Quartus Version 16.1
       PCB design in cooperation with www.GfhR.de
         Copyright (C) by Reinhard Heuberger
          www.pdp11gy.com  info@pdp11gy.com

 Construct RL01/RL02 cartridge format in RAM

*********************************************************
     Clone DL0-RAM area to: DL1:  DL2:  DL3:
     Dump RAM to SD-Card into file:
+------------------------------------------------------------------------+
|    Unit number: 0  >  Write to file RL02_0-8.DEC and RL02_0-8.DSK    |
+------------------------------------------------------------------------+

+------------------------------------------------------------------------+
|    Unit number: 1  >  Write to file RL02_1-8.DEC and RL02_1-8.DSK    |
+------------------------------------------------------------------------+

+------------------------------------------------------------------------+
|    Unit number: 2  >  Write to file RL02_2-8.DEC and RL02_2-8.DSK    |
+------------------------------------------------------------------------+

+------------------------------------------------------------------------+
|    Unit number: 3  >  Write to file RL02_3-8.DEC and RL02_3-8.DSK    |
+------------------------------------------------------------------------+


  RL cartridges Serial-Numbers(HEX), located in file SN8.TXT
      DL0: 0AF3,07A2
      DL1: not in use
      DL2: 08A2,077D
      DL3: 07D4,07A4

                  selected unit: 0
 Started with operating mode:  0100000010100001
```

5. Example

Assuming, we have a real RL02 disk drive, unit **1** and we want to copy the data from the real RL02 to the emulated RL02 disk drives. First, we have to remove the terminator from the emulator board and cabling the real RL02 to be at the end of the RL-bus with connected RL-bus terminator. The real RL02 disk drive is configured as unit **dl1** and the emulator interface board is configured for RL02 units dl0, dl2 and dl3 : SWITCH 7, 6, 4 = ON , **SWITCH 5 = OFF**. Note: The file RL02_3-8.DSK will be used instead of the .DEC file.

Starting the RL-emulator , the following messages appears on the screen :

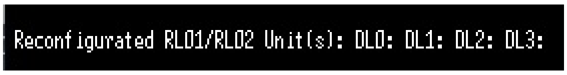


```
      *********>  DEC RL01/RL02 EMULATOR  <*********
        SoC/HPS DE10-Nano board based Version V.2.1
                   (c) WWW.PDP11GY.COM

              >>>>> Device Type = RL02 <<<<
              >>>>>> DEBUG-MODE = ON <<<<<<
              >>>>>> Disk-subset: 8  <<<<<<
     Configurated RL01/RL02 Unit(s): DL0: DL2: DL3:

              ******** ONLINE MODE ********

 ***************************************************************************
                SOC/HPC based V2.1 RL01/RL02 disk emulator
                   developed with Quartus Version 16.1
                 PCB design in cooperation with www.GfhR.de
                 (C) www.pdp11gy.com  info@pdp11gy.com
                           info-file RL8.TXT
           <Edit the file RL8.TXT to change the info-message>
 ***************************************************************************

+----------------------------------------------------------------------------+
|   Unit number: 0 > file RL02_0-8.DEC  used                                 |
+----------------------------------------------------------------------------+

    Unit number: 1  = Not configured

+----------------------------------------------------------------------------+
|   Unit number: 2 > file RL02_2-8.DEC  used                                 |
+----------------------------------------------------------------------------+

+----------------------------------------------------------------------------+
|   Unit number: 3 > file RL02_3-8.DEC  not found, using file RL02_3-8.DSK   |
+----------------------------------------------------------------------------+


  RL cartridges Serial-Numbers(HEX), located in file SN8.TXT
      DL0: 0AF3,07A2
      DL1: not in use
      DL2: 08A2,077D
      DL3: 07D4,07A4

                   selected unit: 0
 Started with operating mode:  0100000010100001
```

Now, we can copy the data from the real RL02 disk drive unit 1 to the emulated RL02 disk drives, for example (RT-11): copy/device dl1: dl0: (dl2: / dl3:)

Here comes a special feature:

- Switch down the real RL02 disk drive
- Set SWITCH 2 = ON (DL2)
- Press button 2 on DE10-Lite board and following message will appear:



```
Reconfigurated RL01/RL02 Unit(s): DL0: DL1: DL2: DL3:
```

From now on, 4 RL02 units will be emulated with full access to the dl2 unit.

Notes:

Customize the disk-set environment: Feel free to modify the File RL8.TXT according to your own needs.

Emulated cartridge SERIAL NUMBER (SN) handling: If file SN8.TXT exist, the content will be used to set the emulated cartridge SERIAL NUMBER. Feel free to modify the File SN8.TXT to change the SERIAL NUMBER.

Example: Convert .DEC file to .DSK file inline with rlemulator and start the PDP-11 emulator using SIMH.

A test RL02 image file is included in the folder socv2_1/ RL/RL02_0-9.DEC **.** It's bootable : RT-11 V05.04 C with Macro-11, BASIC, Fortran, FOCAL + Kermit

Requirement:

- set slide switches to disk-set hex 9 **1-0-0-1**
- configure unit DL0: only, set SW to **0-0-0-1 (Nr. 5 = ON)**
- copy the file socv2_1.zip to DE10-Nano board using scp.
- Extract the zip file:
  root@socfpga:~# **unzip socv2_1.zip**

```
//Steps:

root@socfpga:~# cd socv2_1
root@socfpga:~/socv2_1# ls

!README_V2_1.txt             FW                RL             SIMH
FPGA_configure_from_HPS.jpg  README_V2_0.txt   RL02_0-0.DEC   UTIL

root@socfpga:~/socv2_1# cd RL

root@socfpga:~/socv2_1/RL# chmod 777 *

root@socfpga:~/socv2_1/RL# ls -l
total 11580
-rwxrwxrwx 1 root root      240 Oct 11 17:32 PDP11GY.INF
-rwxrwxrwx 1 root root 11796992 Oct 11 17:37 RL02_0-9.DEC
-rwxrwxrwx 1 root root      498 Oct 11 17:32 RL9.TXT
-rwxrwxrwx 1 root root       36 Oct 11 17:32 SN9.TXT
-rwxrwxrwx 1 root root    43197 Oct 11 17:32 rlemulator

// Note: the .DEC file always has 11796992 byte

// start the rlemulator

root@socfpga:~/socv2_1/RL# ./rlemulator
```

```
      *********>  DEC RL01/RL02 EMULATOR  <*********
        SoC/HPS DE10-Nano board based Version V.2.2
                   (c) WWW.PDP11GY.COM

              >>>>> Device Type = RL02 <<<<
              >>>>>> DEBUG-MODE = ON <<<<<<
              >>>>>> Disk-subset: 9  <<<<<<
     Configurated RL01/RL02 Unit(s): DL0:

              ******** ONLINE MODE ********

 *******************************************************************
               SOC/HPC based V2.2 RL01/RL02 disk emulator
                   developed with Quartus Version 16.1
                 PCB design in cooperation with www.GfhR.de
                 (C) www.pdp11gy.com  info@pdp11gy.com
                           info-file RL9.TXT
           <Edit the file RL9.TXT to change the info-message>

 DL0:  bootable, RT-11 V05.04 C with  Macro-11, BASIC, Fortran, FOCAL
+ Kermit
 DL1:  not configured
 DL2:  not configured
 DL3:  not configured
 *******************************************************************

+...................................................................+
|   Unit number: 0 > file RL02_0-9.DEC  used                        |
+...................................................................+

    Unit number: 1  = Not configured

    Unit number: 2  = Not configured

    Unit number: 3  = Not configured

  RL cartridges Serial-Numbers(HEX), located in file SN9.TXT
      DL0: 0AF3,07A2
      DL1: not in use
      DL2: not in use
      DL3: not in use

                   selected unit: 0
 Started with operating mode:  0100000010100001


//********* Press the RESET Button 1  or force a power fail  *********
//*********************************************************************

        ......... Shutting down system .........
```

```
+-------------------------------------------------------------------+
|    Unit number: 0  >  Write to file RL02_0-9.DEC and RL02_0-9.DSK |
+-------------------------------------------------------------------+

    Unit number: 1  not configured, will be skipped

    Unit number: 2  not configured, will be skipped

    Unit number: 3  not configured, will be skipped

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


// the .DSK file is now available, always 10485760 byte

root@socfpga:~/socv2_1/RL# ls -l
total 21820
-rwxrwxrwx 1 root root      240 Oct 11 17:32 PDP11GY.INF
-rwxrwxrwx 1 root root 11796992 Oct 11 17:37 RL02_0-9.DEC
-rw-r--r-- 1 root root 10485760 Oct 11 17:42 RL02_0-9.DSK
-rwxrwxrwx 1 root root      498 Oct 11 17:32 RL9.TXT
-rwxrwxrwx 1 root root       36 Oct 11 17:32 SN9.TXT
-rwxrwxrwx 1 root root    43197 Oct 11 17:32 rlemulator

// Start the PDP-11 simulator

root@socfpga:~/socv2_1# cd SIMH
root@socfpga:~/socv2_1/SIMH# chmod 777 *
root@socfpga:~/socv2_1/SIMH# ./pdp11

PDP-11 simulator V3.9-0
sim> set CPU 11/23 512k
Disabling CR
Disabling RK
Disabling HK
Disabling TM
sim> attach RL0 ../RL/RL02_0-9.DSK
sim> boot RL0

RT-11SJ  V05.04 C

.SET USR NOSWAP

.SET TT SCOPE

.SET EDIT KED

.INIT/NOQUE VM:
```

```
.dir

DL0DL0.INF    59                 SWAP  .SYS    27P 02-Sep-87
RT11SJ.SYS    79P 15-Jan-88      DD    .SYS     5P 02-Sep-87
DY    .SYS     4P 02-Sep-87      LS    .SYS     5P 02-Sep-87
SL    .SYS    17P 02-Sep-87      TT    .SYS     2P 02-Sep-87
VM    .SYS     3P 02-Sep-87      DU    .SYS     8P 02-Sep-87
LD    .SYS     8P 02-Sep-87      DL    .SYS     5  17-Oct-84
STARTS.COM     1  28-Mar-99      DIR   .SAV    19  02-Sep-87
PIP   .SAV    30  02-Sep-87      DUP   .SAV    49  02-Sep-87
RESORC.SAV    25  02-Sep-87      KED   .SAV    58  02-Sep-87
UCL   .SAV    16  02-Sep-87      CREF  .SAV     6  02-Sep-87
SRCCOM.SAV    26  02-Sep-87      BASIC .SAV    53  04-Apr-83
MACRO .SAV    61  02-Sep-87      DUMP  .SAV     9  02-Sep-87
MKDL0 .BAS     1                 MKDL2 .BAS     1
MKDL3 .BAS     1                 MKDL1 .BAS     1
SYSLIB.OBJ   216  24-Mar-87      FORTRA.SAV   206  24-Mar-87
KERMIT.INI     1  28-Jul-87      KERMIT.SAV   182  02-Apr-86
KERMIT.HLP   148  13-Apr-86      FOCAL .SAV    36  30-Nov-84
FOCALD.SAV    38  30-Nov-84
 35 Files, 1406 Blocks
 18976 Free blocks


// Let’s start BASIC

.r basic
BASIC-11/RT-11 V2.1
OPTIONAL FUNCTIONS (ALL, NONE, OR INDIVIDUAL)? ALL

READY

// Try to run this small program:

10 FOR E=1 TO 30 STEP .1
20 Y=INT((SIN(E)*20)+30)
40 PRINT TAB(Y);"HALLO-1980"
100 NEXT E
```

For comments and questions, please contact me.
INFO@pdp11gy.com

References:

https://github.com/pdp11gy/SoC-HPS-based-RL-disk-emulator
https://github.com/pdp11gy/DEC-RL02-RL01-disk-emulator
http://www.pdp11gy.com/doneE.html