



DIEGO'S HP-41 MODULES

AGENDA

Table of Content

- Introduction
- Reference
- Overview
- Configuration
- Programming
- NoV Runtime Configuration
- HEPAX RAM Clearing
- Closing Topics

INTRODUCTION

Table of Content

- Overview
- Notes
- Acknowledgements
- Goals

Overview

- This presentation is an introduction to Diego Díaz HP-41 modules.
- The modules can be divided into three categories:
 - Clonix modules allows to you to replace several modules by loading their ROM images into one module.
 - NoV modules emulate the Advanced HEPAX (16K) module and the HEPAX Double Memory (16K) module.
Also allows you to load several ROM images into the module.
 - USB-41 module emulate an HP-82143A thermal printer using a USB interface connected to a Windows application.
Also allows you to load several ROM images into the module.
- More information is available at: www.clonix41.org

Notes

- This presentation (105 slides, 40m) is subset of a bigger one (163 slides, 2h) that was planned to be given at HHC2020. More information and the presentation file can be found at these links:
 - Museum thread: www.hpmuseum.org/forum/thread-15460.html
 - Museum article: www.hpmuseum.org/forum/thread-15459.html
 - HHC presentation: www.hhcworld.com/files/hhc2020/sc-hhc2020-p.pdf
 - HPCC presentation ..: www.hhcworld.com/files/hhc2020/sc-hpcc2020-p.pdf
- All underlined text in this presentation represents a link to another slide, a web page or a web downloadable document.
- In September 2020, Diego updated some of his hardware & software, however, this presentation is mostly based on what was available before that update.

Acknowledgements

- **Diego Díaz ...**
 - *for having created these fantastic modules.*
 - *for having created an easy to use application to configure them.*
 - *for his invaluable inputs in making this presentation more accessible.*
 - *for his patience & support.*
- **Monte Dalrymple ...**
 - *for his dedication in keeping updated his HP-41 ROM's archive.*
 - *for his work on his 41CL and on his new add-on modules project. ([41CL Home](#))*
 - *for his invaluable inputs in making this presentation more accessible.*
- **Ángel Martín ...**
 - *for keeping the HP-41C alive by creating mind-blowing ROMs.*
 - *for his invaluable inputs in making this presentation more accessible.*
- **Robert Prosperi ...**
 - *for his invaluable inputs in making this presentation more accessible.*
- **HP-41C users ...**
 - *for still using the HP-41C and keeping it alive.*
 - *for buying Diego's modules allowing him to improve and release new versions.*

Goals

- In this presentation we will ...
 - Review some key informations about the HP-41C system.
 - Discover Diego Díaz modules.
 - Cover every options of three modules. (*Clonix-D, NoV-64d & USB-41*)
 - Go through the programming process.
 - Go through the NoV runtime configuration.
 - Clear NoV HEPAX RAM content.
 - Transfer QROM page

REFERENCE

.....

Table of Content

- [Memory Types](#)
- [HP-41 ROM Words](#)
- [HP-41 RAM Registers](#)
- [HP-41 ROM Memory Map](#)
- [EPROM ROM File Format](#)
- [ERAMCO ROM File Format](#)
- [HEPAX ROM File Format](#)
- [Padded ROM File Format](#)
- [MOD ROM Format](#)
- [MOD File Format](#)
- [MOD File Example](#)
- [LIF File Header](#)
- [HEPAX 4K RAM Structure](#)

Memory Types

| Type | Name | Content at power lost | Erase | Write | Used in ... | Comment |
|--------|----------------------|----------------------------|---|---|--|--|
| RAM | Random Access Memory | Erased | in-circuit (cell level) | in-circuit (cell level) | n/a | This is a family type. RAM Technologies includes: SRAM, DRAM, SDRAM, RDRAM, FRAM & others. |
| SRAM | Static RAM | Erased | in-circuit (cell level) | in-circuit (cell level) | Box: MLDL, RAMBOX Mod: HEPAX | |
| FRAM | Ferroelectric RAM | Preserved | in-circuit (cell level) | in-circuit (cell level) | Box: Mod: NoV | |
| QROM | Quasi-ROM | Follow RAM technology used | in-circuit (word level) | in-circuit (word level) | Box: MLDL, RAMBOX Mod: HEPAX, NoV | Virtual type, build with one of RAM technology. HP-41C ROM cell size is 10 bits word QROM implements ROM word with RAM |
| ROM | Read Only Memory | Preserved | n/a | at-creation (chip level) | Box: Mod: HP, CMT-20 | |
| PROM | Programmable ROM | Preserved | n/a | out-of-circuit with a programmer (chip level) | Box: Mod: CMT-10 | Also called OTP (One Time Programmable) |
| EPROM | Erasable PROM | Preserved | out-of-circuit ultraviolet light (chip level) | out-of-circuit with a programmer (chip level) | Box: MLDL, ROMBOX Mod: CMT-10, ZEPROM | |
| EEPROM | Electrically EPROM | Preserved | in-circuit (cell level) | in-circuit (cell level) | Box: Mod: | |
| Flash | Flash | Preserved | in-circuit (block level) | in-circuit (cell level) | Box: MLDL2000 Mod: NoV, Clonix | Two types of Flash: NOR (memory mapped, same usage as EEPROM) NAND (used for mass storage) |

word = 10 bits / cell = generally one byte / block = 2^n bytes (ex.: 256 or 512 bytes) / chip = entire space of the integrated circuit (IC)
 in-circuit : IC soldered on PCB or inserted in a socket / out-of-circuit : IC unsoldered from the PCB or removed from its socket

HP-41C ROM Words

- Access Type: direct
- Value Size: 10 bit words
- Addressing: 16 bit (4 bits for page & 12 bits for code => 16 pages of 4K word)
- Bank Switching: yes (4 banks for each 4K page)
- Notes:
 - ROM can also be PROM, EPROM, EEPROM or QROM.
 - A ROM page can be either statically or dynamically assigned to a page.
 - Static page: the 4K printer ROM is hardwired to page 6 even though the printer is plugged in one of the 4 ports at the back of the calculator.
 - Dynamic page: a standard 4K application module inserted into port 1 will normally be either mapped to page 8 or page 9.
 - Currently, on Diego's modules, when a bank switch occurs all pages within the module switches to the selected bank not just the page where switch was requested. Diego is working on a new firmware that will allow bank switch to work at the page level instead of at module level.

HP-41C RAM Registers

- Access Type: peripheral
- Value Size: 56 bit registers (7 bytes)
- Addressing: 12 bit (4096 registers addressable but only 1024 available)
- Bank Switching: no
- Notes:
 - 41C peripheral types:
 - RAM Registers, Display, Printer, Card-Reader, Wand, Time, HP-IL, etc.
 - Håkan Thörngren has modified the 41OS to access the full range of RAM addressable registers, but in order to use it you need an hardware device allowing 41OS replacement like the 41CL board or the MLDL2000 unit.
Unfortunately none of Diego's modules has that capability.
Alternative HP-41CL mainframe (OS ROMs) : www.hpmuseum.org/forum/thread-13729.html
 - Ángel Martin has created for the 41CL several modules (CLMEM, SandMatrix, etc.) that uses either some or all of the addressable registers.
Documents: systemyde.com/hp41/documents.html & ROM Images: systemyde.com/hp41/archive.html

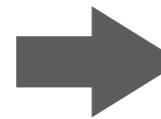
HP-41C ROM Memory Map

.....

| Page | Bank 1 | Bank 2 | Bank 3 | Bank 4 | Note |
|------|--|-----------------------|--------|--------|---------------------------------------|
| #0 | NUT OS 0 ROM | N/A | N/A | N/A | Used by 41C/CV/CX OS |
| #1 | NUT OS 1 ROM | N/A | N/A | N/A | Used by 41C/CV/CX OS |
| #2 | NUT OS 2 ROM | N/A | N/A | N/A | Used by 41C/CV/CX OS |
| #3 | X-Functions ROM (CX) | N/A | N/A | N/A | Avail. for 41C/CV Used by 41CX OS |
| #4 | Disabled HP-IL Printer ROM, Diagnostic ROMs, Lib4 ROM | 41CL Lib4 ROM | | | Takeover & System ROMs |
| #5 | Time ROM, CX Time ROM | CX Ext. Functions ROM | | | CAT 2 - Start Page Order: #5..#F & #3 |
| #6 | Printer ROM | | | | |
| #7 | HP-IL ROM | | | | |
| #8 | | | | | Port 1 - Low |
| #9 | | | | | Port 1 - High |
| #A | | | | | Port 2 - Low |
| #B | | | | | Port 2 - High |
| #C | | | | | Port 3 - Low |
| #D | | | | | Port 3 - High |
| #E | | | | | Port 4 - Low |
| #F | | | | | Port 4 - High |

EPROM ROM File Format

| ADD | HEX | U2:L8 | U2 | L8 |
|------|-----|-------|-------------|----|
| X000 | 2DC | 2:DC | 10 11011100 | |
| X001 | 11E | 1:1E | 01 00011110 | |
| X002 | 0A4 | 0:A4 | 00 10100100 | |
| X003 | 3A1 | 3:A1 | 11 10100001 | |



| Offset | 4xU2 |
|--------|----------|
| 0000 | 11000110 |
| pos: | 03020100 |



| Offset | L8 |
|--------|----------|
| 0000 | 11011100 |
| 0001 | 00011110 |
| 0002 | 10100100 |
| 0003 | 10100001 |



- Format used by EPROM boxes and emulators.
- The ten bits word value is broken into two bits upper and eight bits lower.
- The upper two bits is merged with three other upper two bits values to create an eight bits value. All merged two bits values are save in an U2 file.
- All lower eight bits values are save in one or multiple L8 files.
- The U2 and L8 files get written into two or more EPROMs. (*dictated by the box used*)
- Example: for a 4 K (4096 x 10 bits) ROM, these two EPROM would be needed
 - 2708 for the U2 file. (8192 bits = 4096 x 2 bits = 1024 x 8 bits)
 - 2732 for the L8 file. (32768 bits = 4096 x 8 bits)

ERAMCO ROM File Format

The diagram illustrates the conversion of ERAMCO ROM file format. On the left, there is a table with four rows, each representing a word at a specific address (ADD). The columns are ADD, HEX, and BINARY. The BINARY column shows four 10-bit words: 1011011100, 0100011110, 0010100100, and 1110100001. An arrow points to the right, leading to another table with five rows, each representing a byte at a specific offset. The columns are Offset, HEX, and BINARY. The BINARY column shows five 8-bit bytes: 11000110, 10100001, 10100100, 00011110, and 11011100.

| ADD | HEX | BINARY |
|------|-----|------------|
| X000 | 2DC | 1011011100 |
| X001 | 11E | 0100011110 |
| X002 | 0A4 | 0010100100 |
| X003 | 3A1 | 1110100001 |

| Offset | HEX | BINARY |
|--------|-----|----------|
| 0000 | C6 | 11000110 |
| 0001 | A1 | 10100001 |
| 0002 | A4 | 10100100 |
| 0003 | 1E | 00011110 |
| 0004 | DC | 11011100 |

- Format used by ERAMCO MLDL, ZEPROM and emulators
- Four ten bits words are converted into five bytes.
- A 4096 words ROM is saved into a 5120 bytes file.
- HP-IL mass storage:
 - Save to: ERAMCO SAVEROM & ZEPROM ILSAVE.
 - Read from: ERAMCO GETROM & ZEPROM ILBURN.
- Tools:
 - rom41er convert a padded ROM file into a ERAMCO ROM file with LIF header.
 - er41rom convert a ERAMCO ROM file into a padded ROM file.

HEPAX ROM File Format

The diagram illustrates the conversion process. On the left, a table shows four words of memory in HEPAX format: ADD, HEX, and BINARY. An arrow points to the right, leading to another table where the same data is shown in a standard ROM file format: Offset, HEX, and BINARY.

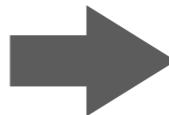
| ADD | HEX | BINARY |
|------|-----|------------|
| X000 | 2DC | 1011011100 |
| X001 | 11E | 0100011110 |
| X002 | 0A4 | 0010100100 |
| X003 | 3A1 | 1110100001 |

| Offset | HEX | BINARY |
|--------|-----|----------|
| 0000 | B7 | 10110111 |
| 0001 | 11 | 00010001 |
| 0002 | E2 | 11100010 |
| 0003 | 93 | 10010011 |
| 0004 | A1 | 10100001 |

- Format used by HEPAX and emulators
- Four ten bits words are converted into five bytes.
- A 4096 words ROM is saved into a 5120 bytes file.
- HP-IL mass storage:
 - Save to: HEPAX WRTROM.
 - Read from: HEPAX READROM.
- Tools:
 - rom41hx convert a padded ROM file into a HEPAX ROM file with LIF header.
 - hx41rom convert a HEPAX ROM file with LIF header into a padded ROM file.

Padded ROM File Format

| ADD | HEX | BINARY |
|------|-----|------------|
| X000 | 2DC | 1011011100 |
| X001 | 11E | 0100011110 |
| X002 | 0A4 | 0010100100 |
| X003 | 3A1 | 1110100001 |



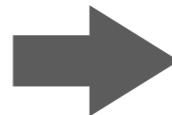
| Offset | HEX | BINARY |
|--------|-----|----------|
| 0000 | 02 | 00000010 |
| 0001 | DC | 11011100 |
| 0002 | 01 | 00000001 |
| 0003 | 1E | 00011110 |
| 0004 | 00 | 00000000 |
| 0005 | A4 | 10100100 |
| 0006 | 03 | 00000011 |
| 0007 | A1 | 10100001 |

- Format used by Clonix & NoV modules, 41CL, MLDL2000, some EPROM boxes and emulators.
- The ten bits word value is left padded with zero's to create a sixteen bits value.
- A 4096 words ROM is saved into a 8192 bytes file.
- 41CL:
 - NEWT processor uses bit 13 & 12 (TT) to manage its turbo feature.
 - 41CL CX system ROMs has been "turbo" modified.
- Tools:
 - rom41lif convert a padded ROM file into a ROM file with LIF header.

| Offset | HEX | BIN |
|--------|-----|----------|
| 0000 | 02 | 00TT0010 |
| 0001 | DC | 11011100 |

MOD ROM Format

| ADD | HEX | BINARY |
|------|-----|------------|
| X000 | 2DC | 1011011100 |
| X001 | 11E | 0100011110 |
| X002 | 0A4 | 0010100100 |
| X003 | 3A1 | 1110100001 |



| Offset | HEX | BINARY |
|--------|-----|----------|
| 0000 | DC | 11011100 |
| 0001 | 7A | 01111010 |
| 0002 | 44 | 01000100 |
| 0003 | 4A | 01001010 |
| 0004 | E8 | 11101000 |

- Format used in MOD file format.
- Four ten bits words are converted into five bytes.
- A 4096 words ROM is converted into a 5120 bytes array.
- Tools:
 - lifmod can export MOD ROM images to padded ROM files.

MOD File Format

- Format used by MLNL2000, DM41X and emulators.
- Can hold up to 255 ROMs.

- Tools:

- lifmod can list the content of a MOD file.
- MLNL2000 GUI can create, read and update a MOD file.

| ModuleFileHeader | ModuleFilePages [0..255] | | |
|------------------|--------------------------|------------------|------------------|
| | ModuleFilePage 0 | ModuleFilePage 1 | ModuleFilePage X |
| 729 Bytes | 5188 Bytes | 5188 Bytes | 5188 Bytes |

```
typedef struct
{
    char FileFormat      [5];
    char Title           [50];
    char Version          [10];
    char PartNumber       [20];
    char Author           [50];
    char Copyright        [100];
    char License          [200];
    char Comments         [255];
    byte Category;
    byte Hardware;
    byte MemModules;
    byte XMemModules;
    byte Original;
    byte AppAutoUpdate;
    byte NumPages;
    byte HeaderCustom     [32];
}
```

| Category | |
|----------|---------------------|
| Value | Description |
| 0 | Undefined |
| 1 | Operating System |
| 2 | Application PAC |
| 3 | HP-IL Peripheral |
| 4 | Standard Peripheral |
| 5 | Custom Peripheral |
| 6 | Beta |
| 7 | Experimental |

| Hardware | |
|----------|--------------------------|
| Value | Description |
| 0 | None |
| 1 | 82143A Printer |
| 2 | 82104A Card Reader |
| 3 | 82182A Time Module |
| 4 | 82153A Barcode Wand |
| 5 | 82160A HP-IL Module |
| 6 | 82242A IR Printer Module |
| 7 | HEPAX Module |
| 8 | W&W RAMBox |
| 9 | MLNL2000 |
| 10 | Clonix/NoV Modules |

| MemModules | |
|------------|------------------|
| Value | Description |
| 0 | No Memory Module |
| 1 | 1 Memory Module |
| 2 | 2 Memory Modules |
| 3 | 3 Memory Modules |
| 4 | 4 Memory Modules |

| Original | |
|----------|-------------|
| Value | Description |
| 0 | Updated |
| 1 | Original |

| Hardware | |
|----------|--------------------------|
| Value | Description |
| 0 | None |
| 1 | 82143A Printer |
| 2 | 82104A Card Reader |
| 3 | 82182A Time Module |
| 4 | 82153A Barcode Wand |
| 5 | 82160A HP-IL Module |
| 6 | 82242A IR Printer Module |
| 7 | HEPAX Module |
| 8 | W&W RAMBox |
| 9 | MLNL2000 |
| 10 | Clonix/NoV Modules |

| XMemModules | |
|-------------|-----------------------|
| Value | Description |
| 0 | None |
| 1 | X-Functions/Memory |
| 2 | XFM + 1 X-Mem Module |
| 3 | XFM + 2 X-Mem Modules |

| AppAutoUpdate | |
|---------------|---------------|
| Value | Description |
| 0 | Do Not Update |
| 1 | Overwrite |

```
typedef struct
{
    char Name           [20];
    char ID             [9];
    byte Page;
    byte PageGroup;
    byte Bank;
    byte BankGroup;
    byte RAM;
    byte WriteProtect;
    byte FAT;
    byte Image          [5120]; // MOD ROM Format
    byte PageCustom     [32];
}
```

ModuleFilePage; // struct size = 5188 bytes

References:

[hp.giesselink.com/v41.htm](http://giesselink.com/v41.htm)

www.hpcalc.org/details/3695

www.hp41.org/LibView.cfm?Command=View&ItemID=1352 (login required)

MOD File Example

| ModuleFileHeader | ModuleFilePages [0..255] | | |
|------------------|--------------------------|------------------|------------------|
| | ModuleFilePage 0 | ModuleFilePage 1 | ModuleFilePage X |
| 729 Bytes | 5188 Bytes | 5188 Bytes | 5188 Bytes |

| | |
|--|---|
| <i>FileName:</i> HPIL.MOD | <i>ModuleFilePage [0]</i> |
| <i>ModuleFileHeader</i> | <i>Name:</i> ILPrinter-2E |
| <i>FileFormat:</i> MOD1 | <i>ID:</i> PL2E |
| <i>Title:</i> HP-IL Module | <i>Page:</i> 6 (must be in this location) |
| <i>Version:</i> EH | <i>PageGroup:</i> 0 (not grouped) |
| <i>PartNumber:</i> 82160A | <i>Bank:</i> 1 |
| <i>Author:</i> Hewlett-Packard | <i>BankGroup:</i> 0 (not grouped) |
| <i>Copyright:</i> Hewlett-Packard | <i>RAM:</i> 0 (no) |
| <i>License:</i> Hewlett-Packard Company makes no warranty as to the accuracy or completeness of the foregoing information and hereby disclaims any responsibility therefore. | <i>WriteProtect:</i> 0 (no or not applicable) |
| <i>Comments:</i> | <i>FAT:</i> 1 (yes) |
| <i>Category:</i> 3 (HP-IL Peripheral) | <i>Image:</i> [IL Printer rom image : 5120 bytes in MOD ROM format] |
| <i>Hardware:</i> 5 (82160A HP-IL Module) | <i>PageCustom:</i> |
| <i>MemModules:</i> 0 (no memory modules) | <i>ModuleFilePage [1]</i> |
| <i>XMemModules:</i> 0 (no extended memory) | <i>Name:</i> ILModule-1H |
| <i>Original:</i> 1 (yes) | <i>ID:</i> CS1H |
| <i>AppAutoUpdate:</i> 0 (no) | <i>Page:</i> 7 (must be in this location) |
| <i>NumPages:</i> 2 | <i>PageGroup:</i> 0 (not grouped) |
| <i>HeaderCustom:</i> | <i>Bank:</i> 1 |
| | <i>BankGroup:</i> 0 (not grouped) |
| | <i>RAM:</i> 0 (no) |
| | <i>WriteProtect:</i> 0 (no or not applicable) |
| | <i>FAT:</i> 1 (yes) |
| | <i>Image:</i> [IL Module rom image : 5120 bytes in MOD ROM format] |
| | <i>PageCustom:</i> |

LIF File Header

- Logical Information Format.
- LIF Header length is 32 bytes.
- Craig A. Finseth's LIF Page.

www.finseth.com/hpdata/lif.php

- Dan McDonald's HP-IL Files.

www.hpmuseum.org/cgi-sys/cgiwrap/hpmuseum/articles.cgi?read=24

- Joachim Siebold's lifutils.

github.com/bug400/lifutils

- Based on works from Tony Duell, Leo Duran, Warren Furlow, Christophe Gottheimer, Heinz W. Werntges & Matrin Kroeker.

| Offset | Length | Description |
|--------|--------|-------------------------------------|
| 0 | 10 | File name |
| 10 | 2 | File type (see table below) |
| 12 | 4 | Start block |
| 16 | 4 | Allocated length in blocks |
| 20 | 6 | Creation date & time [YYMMDDHHMMSS] |
| 26 | 6 | Miscellaneous data |

| Type | Name | Description |
|------|--------|-------------------------------|
| E020 | WAXM41 | HP-41 write all with X-Memory |
| E030 | XM41 | HP-41 write all with X-Memory |
| E040 | ALL41 | HP-41 write all |
| E050 | KEY41 | HP-41 user keys assignment |
| E060 | STAT41 | HP-41 status |
| E070 | X-M41 | HP-41 ROM ERAMCO |
| E080 | PGM41 | HP-41 FOCAL program |
| E0D0 | SDATA | HP-41 data file |

HEPAX 4K RAM Structure

| Addr. | HEX | Comment |
|-------|-----|--|
| X000 | | XROM Number, HEPAX assign an unused XROM ID |
| X001 | 000 | CAT Entries, 00 = none, set to zero by HEPAX |
| X002 | 000 | |
| X... | 000 | FAT Space (64 fn + end-of-fat), unused and set to zero by HEPAX |
| X083 | 000 | |
| X084 | 000 | |
| X... | 000 | Unused by HEPAX |
| X08F | 000 | |
| X090 | | |
| X... | | HEPAX Data |
| XFE5 | | |
| XFE6 | 000 | Unknown, spacer ? |
| XFE7 | 000 | HEPAX pages linked list: previous page (000 = end of list) |
| XFE8 | 000 | HEPAX pages linked list: next page (000 = end of list) |
| XFE9 | 091 | HEPAX first file address |
| XFEA | 000 | 0091 |
| XFEB | 000 | HEPAX active File address |
| XFEC | 000 | 0000 (0000 = None) |
| XFED | 090 | HEPAX usable space start address |
| XFEE | 000 | 0090 |
| XFFF | 091 | HEPAX next file address |
| XFF0 | 000 | 0091 |
| XFF1 | 0E5 | HEPAX usable space end address |
| XFF2 | 00F | 0FE5 |
| XFF3 | 200 | Initial value = 100 and set to 200 after initialization |
| XFF4 | 000 | IVT Pause Loop, unused and set to zero by HEPAX |
| XFF5 | 000 | IVT Main Running Loop, unused and set to zero by HEPAX |
| XFF6 | 000 | IVT Deep Sleep Wake up, no key down, unused and set to zero by HEPAX |
| XFF7 | 000 | IVT Off, unused and set to zero by HEPAX |
| XFF8 | 000 | IVT I/O Service, unused and set to zero by HEPAX |
| XFF9 | 000 | IVT Deep Sleep Wake up, unused and set to zero by HEPAX |
| XFFA | 000 | IVT Cold Start, unused and set to zero by HEPAX |
| XFFB | 000 | |
| XFFC | 000 | ROM Trailer, unused and set to zero by HEPAX |
| XFFD | 000 | |
| XFFE | 000 | |
| XFFF | 000 | Checksum, unused and normally set to zero by HEPAX |

Note: HEPAX RAM structure decoding is a work in progress and may contain invalid information, please use with caution.

OVERVIEW

Table of Content

- Modules
- Software

MODULES

Table of Content

- History
- Timeline
- Specification Table
- Specification Notes
- Resources & Web Links

History

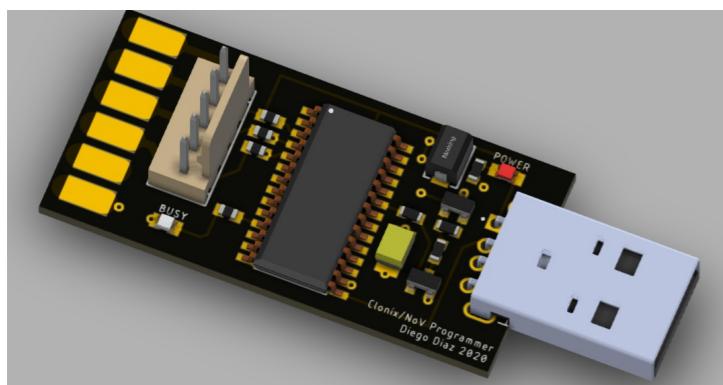
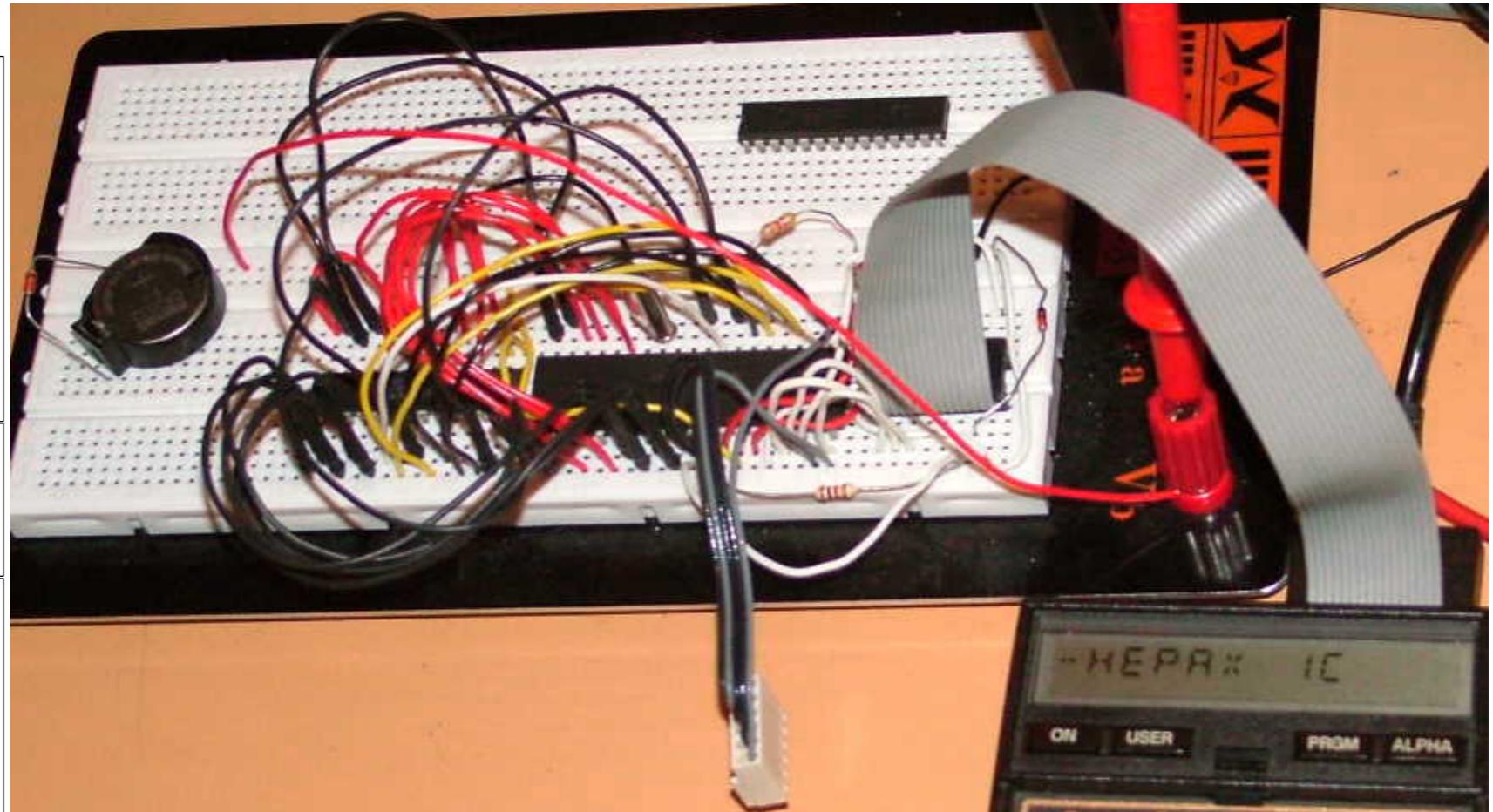
| Date | EPROM Devices | MLDL Devices (RAM/EPROM) |
|----------|----------------------------|----------------------------|
| Nov 1981 | HHP-16K (16K) | |
| Apr 1982 | | MLDL-I (4K) |
| Jun 1982 | ProtoEPROM (4K, 8K, 16K) | ProtoCODER (4K) |
| ??? 1982 | HHP-32K (32K) | |
| Jan 1983 | HP-IL EPROM Programmer | MLDL-II |
| Jun 1983 | ERAMCO (24K) | ESMLDL 1 (8K/24K) |
| Sep 1983 | HHP-PE (32K) | |
| Nov 1983 | | ProtoCODER-2 (4K) |
| Jun 1984 | MBK-16 (16K) | MBK-ProfiSET (16K/8K) |
| Aug 1984 | ERAMCO (32K) | ESMLDL (8K/24K) |
| Dec 1984 | CMT-100 (4K, 8K, 16K) | |
| Aug 1985 | CMT-110 (16K, 32K) | |
| | CMT-10 (4K, 8K, 16K) | |
| Dec 1985 | W&W EPROMBOX (32K) | W&W RAMBOX (32K) |
| Dec 1985 | | ERAMCO RSU1 (16K) |
| Jan 1986 | SOS HP-IL EPROM Programmer | |
| Apr 1988 | ZEPROM (16K) | |
| ??? 1988 | | W&W RAMBOX II (64K) |
| | | ES RAMBOX (32K, 64K, 128K) |
| | | HEPAX (8K) |
| | | HEPAX Memory (8K) |
| | | Adv. HEPAX (16K) |
| | | HEPAX Double Memory (16K) |
| Sep 2005 | | MLDL2000 (512K/2M Flash) |



Main reference: Jeremy Smith, CHHU v2n7p59, Nov. 1985

Timeline

| Date | Module | Event |
|----------|------------------------|------------------|
| Mar 2003 | Clonix | project started |
| Jul 2003 | 1 st ML_DL | built (7) |
| Sep 2003 | 1 st Module | built (8) |
| Dec 2003 | Clonix 41 | released (1 & 2) |
| Mar 2004 | NoVRAM | project started |
| Jul 2004 | NoVRAM | released (1 & 3) |
| Oct 2005 | NoV-32 | released (1 & 4) |
| May 2008 | Clonix-D | released (1 & 5) |
| Sep 2008 | NoV-64 | released (1 & 5) |
| Apr 2012 | USB-41 | released (1 & 6) |
| Dec 2013 | Clonix 41 | discontinued |
| | NoVRAM | discontinued |
| | NoV-32 | discontinued |
| | NoV-64 | discontinued |
| | NoV-64d | released (1 & 5) |
| | Clonix 41d | Anniversary Ed. |
| Sep 2020 | USB-41 | re-released |
| Sep 2020 | PICkit 2 | released |



Specifications Table

| Features \ Modules | Clonix 41 silver | Clonix 41 gold | Clonix 41d gold | Clonix-D * | USB-41 | USB-41 * | NoVRAM | NoV-32 | NoV-64 | NoV-64d * |
|--|-------------------|-------------------|-------------------|-----------------|-------------------|------------------|-------------------|-------------------|-------------------------|-------------------------|
| Released ^ Discontinued | 2003-12 ^ 2013-12 | 2003-12 ^ 2013-12 | 2013-12 ^ 2013-12 | 2008-05 ^ Now | 2012-04 ^ 2020-09 | 2020-09 ^ Now | 2004-07 ^ 2013-12 | 2005-10 ^ 2013-12 | 2008-09 ^ 2013-12 | 2013-12 ^ Now |
| Microcontroller | PIC18LF252 | PIC18LF252 | PIC18LF252 | PIC18LF2620 | PIC18LF2620 | PIC18LF2620 | PIC18LF252 | PIC18LF252 | PIC18LF2620 | PIC18LF2620 |
| ROM size | 24K words | 24K words | 24K words | 48K words | 48K words | 48K words | 24K words | 24K words | 48K words | 48K words |
| ROM pages | 6 | 6 | 6 | 6 | 12 | 12 | 6 | 6 | 6 | 6 |
| ROM blocks | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 2 | 2 |
| ROM blocks mergeable | — | — | — | ✓ | ✗ | ✗ | — | — | ✓ | ✓ |
| ROM hard preload | none | none | none | none | 82143A (4x4K) | 82143A (4x4K) | HEPAX (4x4K) | HEPAX (4x4K) | HEPAX (4x4K) | HEPAX (4x4K) |
| ROM pages available | 6 | 6 | 6 | 12 | 8 (12-4) | 8 (12-4) | 2 (6-4) | 2 (6-4) | 8 (12-4) | 8 (12-4) |
| ROM block select ^{(a)(b)} | — | — | Port Sensing | Port Sensing | — | — | — | — | Control Word | Control Word |
| RAM size | 512 words | 512 words | 512 words | — | — | — | 16K words | 32K words | 64K words | 64K words |
| RAM pages ^(c) | 0.125 or 1/8 | 0.125 or 1/8 | 0.125 or 1/8 | — | — | — | 4 | 4 | 4 | 4 |
| RAM blocks | 1 | 1 | 1 | — | — | — | 1 | 2 | 4 | 4 |
| RAM block select ^(b) | — | — | — | — | — | — | — | Control Word | Control Word | Control Word |
| RAM type ^(d) | SRAM | SRAM | SRAM | — | — | — | FRAM | FRAM | FRAM | FRAM |
| Page shadowing ^(e) | ROM | ROM | ROM | ROM | ROM | ROM | ROM & RAM | ROM & RAM | ROM & RAM | ROM & RAM |
| HEPAX support | ✓ (ROM only) | ✓ (ROM only) | ✓ (ROM only) | — | — | — | ✓ | ✓ | ✓ | ✓ |
| HEPAX ROM relocation | ✗ | ✗ | ✗ | — | — | — | ✓ | ✓ | ✓ | ✓ |
| HEPAX RAM protection | — | — | — | — | — | — | ✓ | ✓ | ✓ | ✓ |
| HEPAX RAM max mapped | — | — | — | — | — | — | 16K | 16K | 32K | 32K |
| 41C 1.7x turbo mode ^(f) | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| 41CL compatible | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Bank switching (4 banks) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Diagnostic modules | single | single | single & double | single & double | ✗ | ✗ | single | single | single | single |
| Advantage module | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ |
| Forth-41 | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| Double HEPAX RAM | — | — | — | — | — | — | ✓ | ✓ | ✓ | ✓ |
| Double X-Memory | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ |
| W&W HP-41CY / RAMBox64 | — | — | — | — | — | — | ✗ | ✗ | ✓ | ✓ |
| Page transfer to/from PC | ✗ | ✗ | ✗ | ✗ | ✓ ^(g) | ✓ ^(g) | ✗ | ✗ | ✓ | ✓ |
| USB/RS-232 chip | — | — | — | — | Prolific PL2303 | FTDI FT232BL | — | — | — | — |
| Alternate persona ^(h) | — | — | — ⁽ⁱ⁾ | — | — | — | Clonix 41 | Clonix 41 | Clonix-D ^(j) | Clonix-D ^(k) |
| Pwr: sleep ^{uA} \standby ^{uA} \run ^{mA} | 10 \ 100 \ 9.5 | 10 \ 100 \ 13.5 | 10 \ 100 \ 13.5 | — | — | — | — | — | — | — |
| Module price: | — | — | — | 100.00 € | — | 110.00 € | — | — | — | 140.00 € |
| PICkit2 Programmer price: | — | — | — | 40.00 € | — | 40.00 € | — | — | — | 40.00 € |

Legend: [✓ : yes] [✗ : no] [— : n/a]

Clonix 41

Clonix 41

Clonix 41d

Clonix-D

USB-41

USB-41

NoVRAM

NoV-32

NoV-64

NoV-64d

* Available at www.clonix41.org

Specifications Table

| Features \ Modules | Clonix-D * | NoV-64d * | USB-41 * |
|------------------------------------|--------------|-------------------------|------------------|
| ROM size | 48K words | 48K words | 48K words |
| ROM pages | 6 | 6 | 12 |
| ROM blocks | 2 | 2 | 1 |
| ROM blocks mergeable | ✓ | ✓ | ✗ |
| ROM preload | none | HEPAX (4x4K) | 82143A (4x4K) |
| ROM pages available | 12 | 8 (12-4) | 8 (12-4) |
| ROM block select ^{(a)(b)} | Port Sensing | Control Word | — |
| RAM size | — | 64K words | — |
| RAM pages ^(c) | — | 4 | — |
| RAM blocks | — | 4 | — |
| RAM block select ^(b) | — | Control Word | — |
| RAM type ^(d) | — | FRAM | — |
| Page shadowing ^(e) | ROM | ROM & RAM | ROM |
| HEPAX support | — | ✓ | — |
| HEPAX ROM relocation | — | ✓ | — |
| HEPAX RAM protection | — | ✓ | — |
| HEPAX RAM max mapped | — | 32K | — |
| Double X-Memory | ✓ | ✓ | ✗ |
| W&W HP-41CY / RAMBox64 | — | ✓ | — |
| Page transfer to/from PC | ✗ | ✓ | ✓ ^(g) |
| Alternate persona ^(h) | — | Clonix-D ^(k) | — |
| Module price: | 100.00 € | 140.00 € | 110.00 € |
| PICkit2 Programmer price: | 40.00 € | 40.00 € | 40.00 € |

Legend: [✓ : yes] [✗ : no] [— : n/a]



* Available at www.clonix41.org

Specifications Notes

- a. A port sensing module is able to select a Flash block based on its plugged location (odd or even port).
- b. Control word allows to choose which RAM and/or Flash block is mapped into the HP-41C memory space.
- c. 4K RAM pages can be configured as HEPAX RAM (default) or QROM.
- d. When module is unplugged SRAM content is lost while FRAM content is preserved.
- e. Page shadowing allows a physical module to take precedence over a Clonix or NoV module mapped page.
- f. When configured with the **Standard 6P** option, the module is able to work in a speed up HP-41C (1.7X turbo hardware upgrade).
- g. USB-41 can transfer an HP-41 ROM to a Padded ROM file on a PC but need a RAMBox/MLDL/NoV unit to transfer a Padded ROM file from a PC to a QROM space on the HP-41.
- h. NoV modules can be programmed and act as a Clonix module.
- i. Clonix 41d Anniversary Edition was delivered pre-loaded with Service Module 1C (C/CV) active in even ports and Service Module 2A (CV/CX) active in odd ports.
- j. NoV-64 lack the port sensing hardware of the Clonix-D module.
- k. NoV-64d is in fact two modules in one. It can be either configured as a NoV-64 module or as a full Clonix-D module.

Resources and web links

► Diego Díaz Projects

1. Clonix & NoV Configuration Utility (v4.2)
www.clonix41.org/Projects/Updates/Clonix_CD_090315.zip
2. Clonix 41 Project Page & Manual
www.clonix41.org/Projects/Clonix-41/Cloni41_00.htm
www.clonix41.org/Projects/Clonix-41/clonix_man.zip
3. NoVRAM Project Page & Support Files
www.clonix41.org/Projects/Novram/Novram_00.htm
www.clonix41.org/Projects/Novram/novram-hepax.zip
4. NoV-32 Project Page, Manual & Support Files
www.clonix41.org/Projects/Nov32/Nov32_00.htm
www.clonix41.org/Projects/Nov32/New_HW.htm
www.clonix41.org/Projects/Nov32/NoV-32_Usr-man.pdf
www.clonix41.org/Projects/Nov32/NoV-32_SW.zip
5. NoV-64 Project Page & Manual
www.clonix41.org/Projects/Nov64/Nov64_00.htm
www.clonix41.org/Projects/Nov64/NoV-64v08r_Man3.pdf
6. USB-41 Application, Manual & Support Files
www.clonix41.org/Projects/USB-82143A/USB-82143A.zip
www.clonix41.org/Projects/USB-41/USB-41-4.rar

ClonixConfig (v6.1) & Utilities Update (2020-09)
www.dropbox.com/s/ene5dbwq3r8qr14/Updt_2020-09.zip

Other Projects & Files

Modules Chart : www.clonix41.org/Projects/Clonix-NoV_chart.pdf
I/O Block : www.clonix41.org/Maintenance/IO_Block/IO_Block.htm

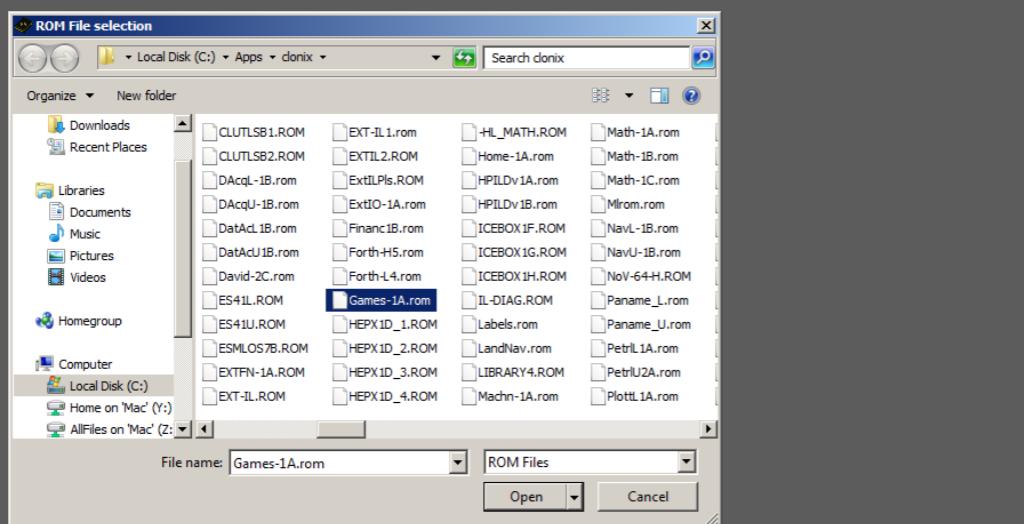
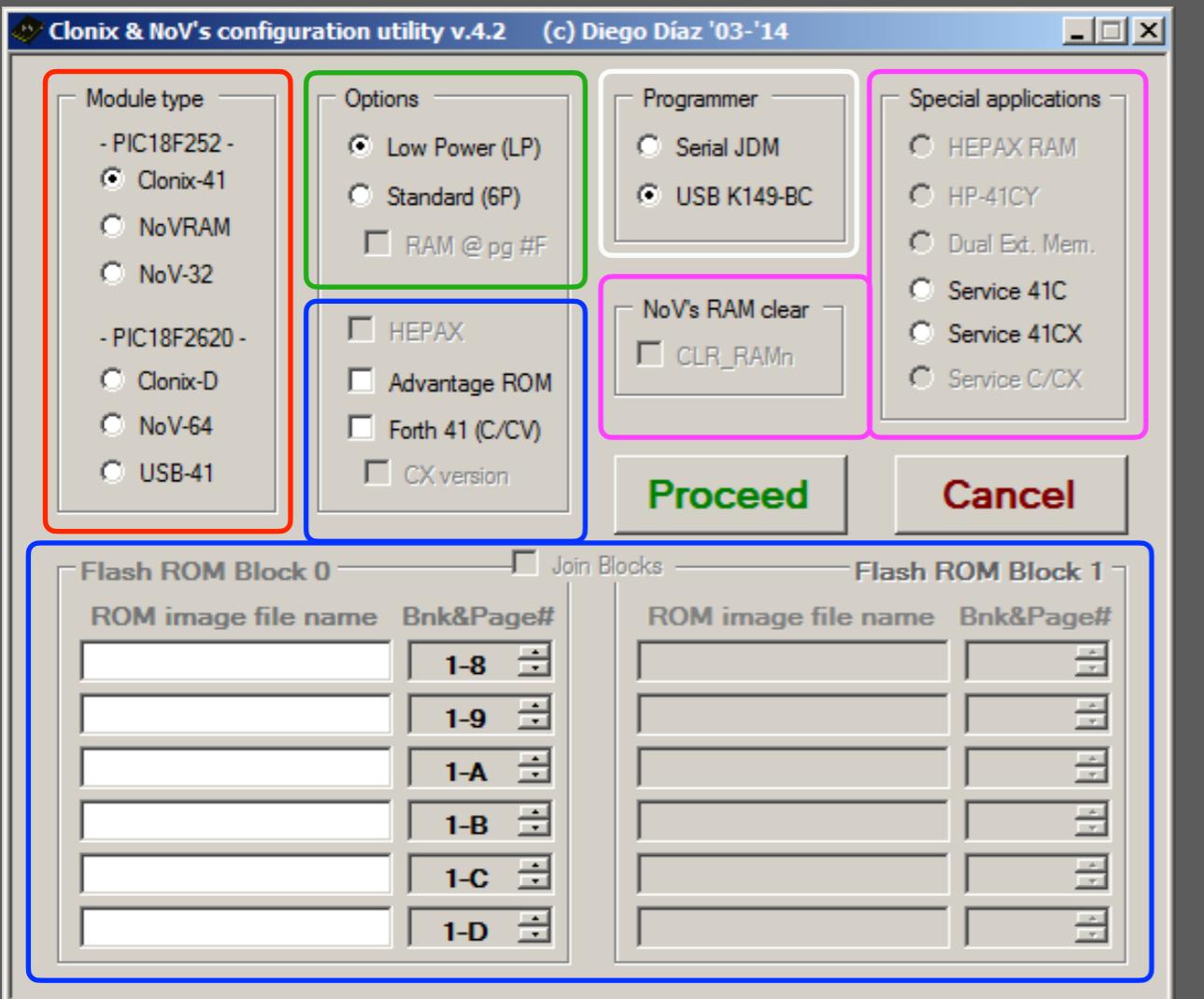
► Other Projects & Web Sites

7. Based on Lynn A. Wilkins design, developer of the first Machine Language Development Lab. PPC Journal V9N3P27 (PAHHC Jake Schwartz)
www.pahhc.org/ppccdrom.htm
 8. Using the ROM-PAC emulation code for PIC18C252 written by John Ioannidis.
archived.hpcalc.org/museumforum/thread-9845.html
 9. HHP-16K EPROM Emulator Introduction
www.embeddedcomponents.com/blogs/2007/09/introduction-to-hhp-16k-eprom-emulator/
 10. MLDL2000 (Meindert Kuiprs)
hp41.kuiprs.nl/hp41.htm
- HP-41C Dedicated Site (Warren Furlow)
www.hp41.org
- HP-41C ROM Images (Monte Dalrymple)
systemyde.com/hp41/archive.html
- HP Calculators Museum (David Hicks)
www.hpmuseum.org
www.hpmuseum.org/forum/index.php
- HP Calc. Museum Archives (Eric Rechlin)
archived.hpcalc.org/museumforum/
- Silicium Forum (French Site, Hand Held Section)
<http://www.silicium.org/forum/viewforum.php?f=46>

SOFTWARE

Table of Content

- [Configuration Utility v4.2](#)
- [Configuration Utility v6.1](#)
- [MPASM](#)
- [MicroBurn DIY K150](#)
- [PICKit 2 Programmer](#)
- [Windows Device Manager](#)
- [USB 82143A](#)
- [USB-41 Page Transfer](#)

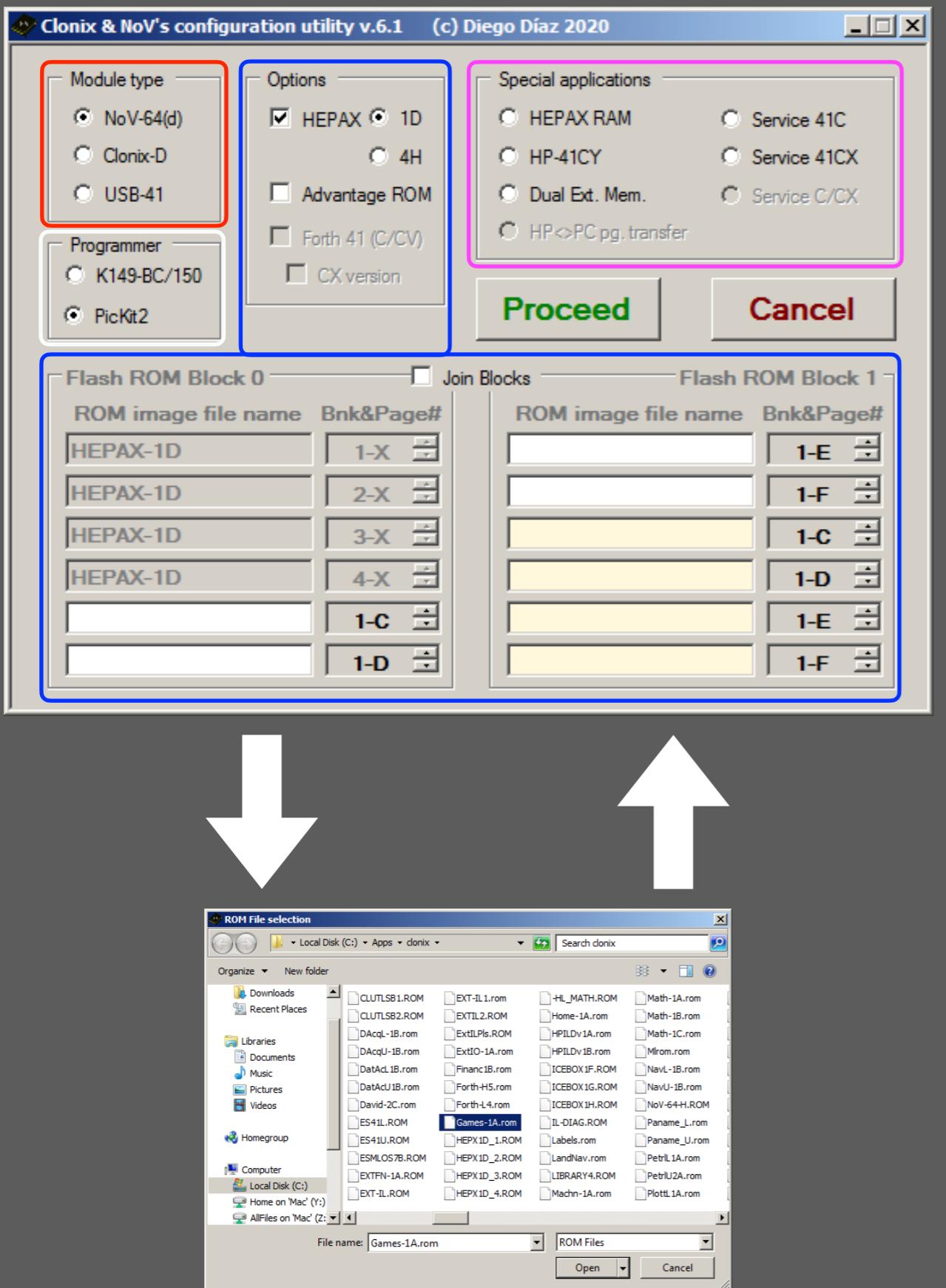


Configuration Utility v4.2

- Windows application for configuring Clonix and NoV modules based on PIC18F2620 or PIC18F252.
- Module Type group (red) is where you select your module.
- Options group (green) is specific to the Clonix-41 module.
- Options and Flash ROM groups (blue) is where you select pre-configured ROM images and/or manually loaded ROM images.

Pink and Blue+Green rectangles options are mutually exclusive.

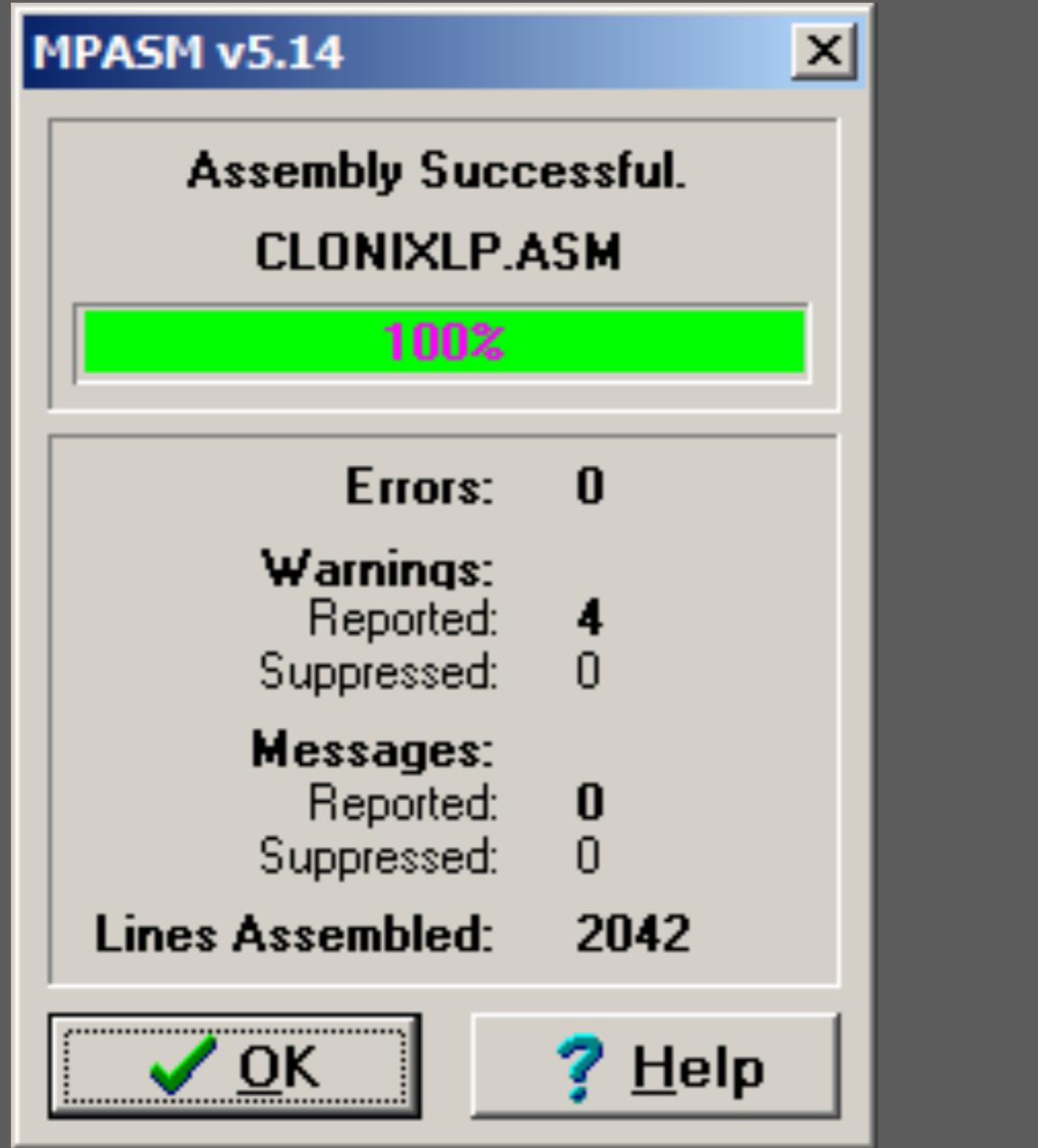
- Programmer group (white) is where you select which type of PIC programmer you are using: RS-232 or USB.
- Proceed button
 - generate an assembly file.
 - call the Microchip PIC assembler that compile the assembly source code and generate an Intel hex file.
 - call the PIC programming software to transfer the hex file to the module.
- Cancel button exit the application.



Configuration Utility v6.1

.....

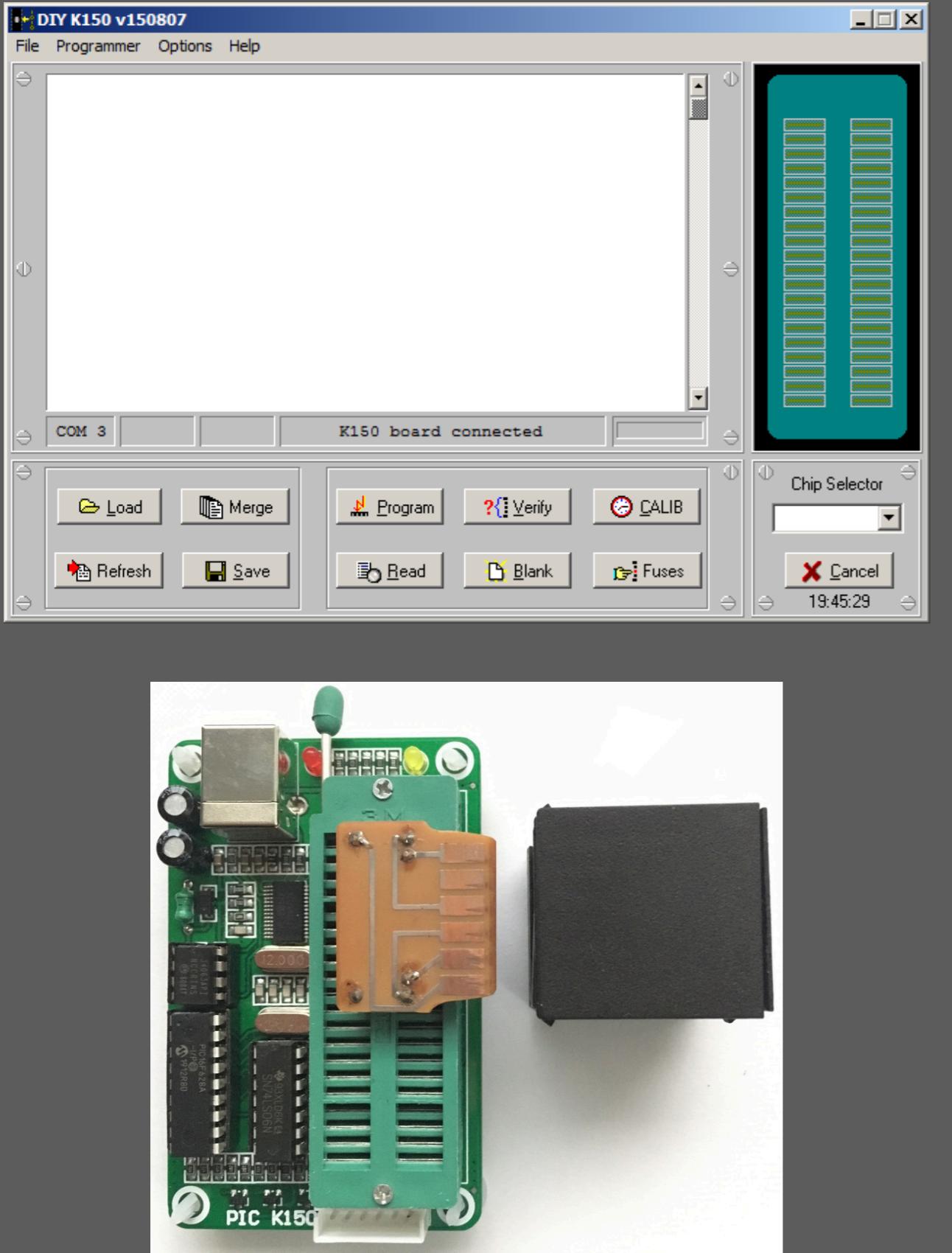
- Windows application for configuring Clonix and NoV modules based on PIC18F2620 or later.
 - Module Type group (*red*) is where you select your module.
 - Options and Flash ROM groups (*blue*) is where you select pre-configured ROM images and/or manually loaded ROM images.
 - ROM File Selection dialog box appears each time you click in one of the ROM image file name text boxes.
 - Special applications and NoV's RAM clear groups (*pink*) is where you select an atypical functionality.
- Pink and Blue rectangles options are mutually exclusive.*
- Programmer group (*white*) is where you select which type of USB PIC programmer you are using.
 - Proceed button
 - generate an assembly file.
 - call the Microchip PIC assembler that compile the assembly source code and generate an Intel hex file.
 - call the PIC programming software to transfer the hex file to the module.
 - Cancel button exit the application.



MPASM

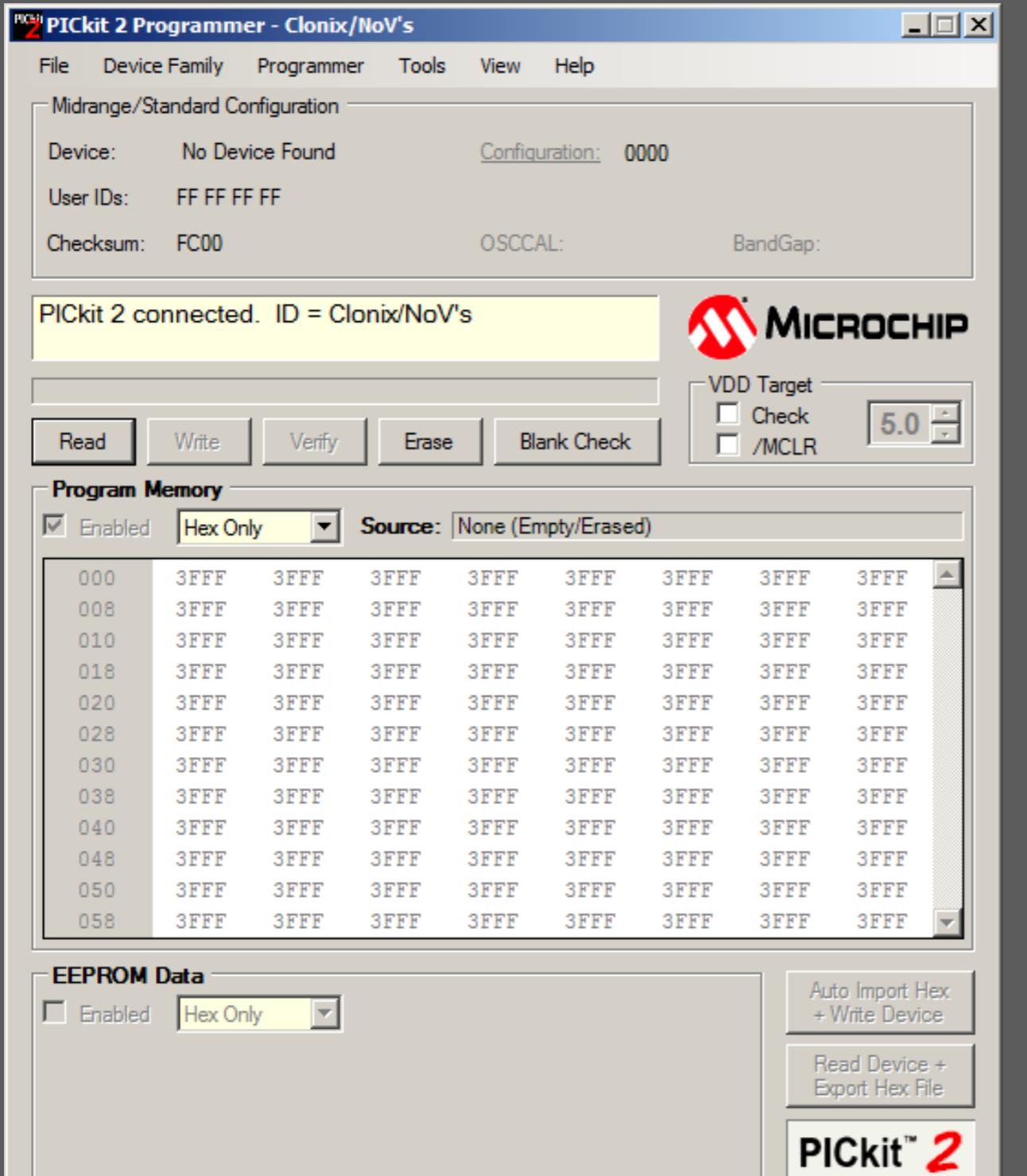
.....

- Assembler made by Microchip for their Peripheral Interface Controller (PIC) and called by Clonix & NoV Configuration Utility.
- MPASM takes the assembly file created by Clonix & NoV Configuration Utility, generate an executable binary file for the PIC microcontroller then serialize it as an extended Intel HEX file format.
- Included in Clonix & NoV Configuration Utility v4.2 and v6.1 packages.



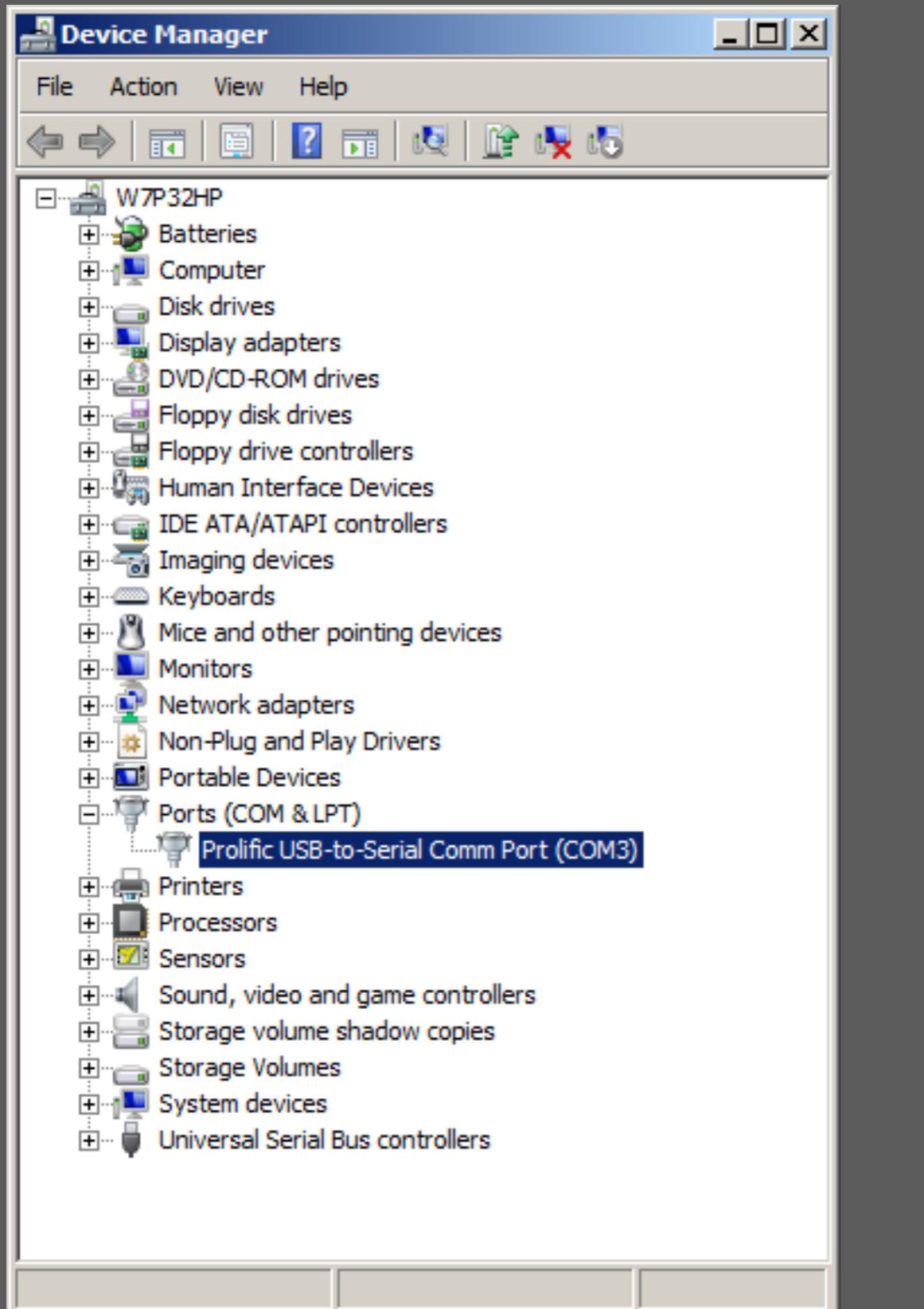
MicroBurn DIY K150

- The purpose of this application is to upload/download a PIC Extended Intel HEX file to/from the microcontroller.
- MicroBrn is a 32 bits Windows application co-developed by DIY Electronics and by Jim Robertson of Newfound Electronics.
- MicroBrn last release is August 2007
- Available at www.kitsrus.com
- Included in Clonix & NoV Configuration Utility v4.2 package.



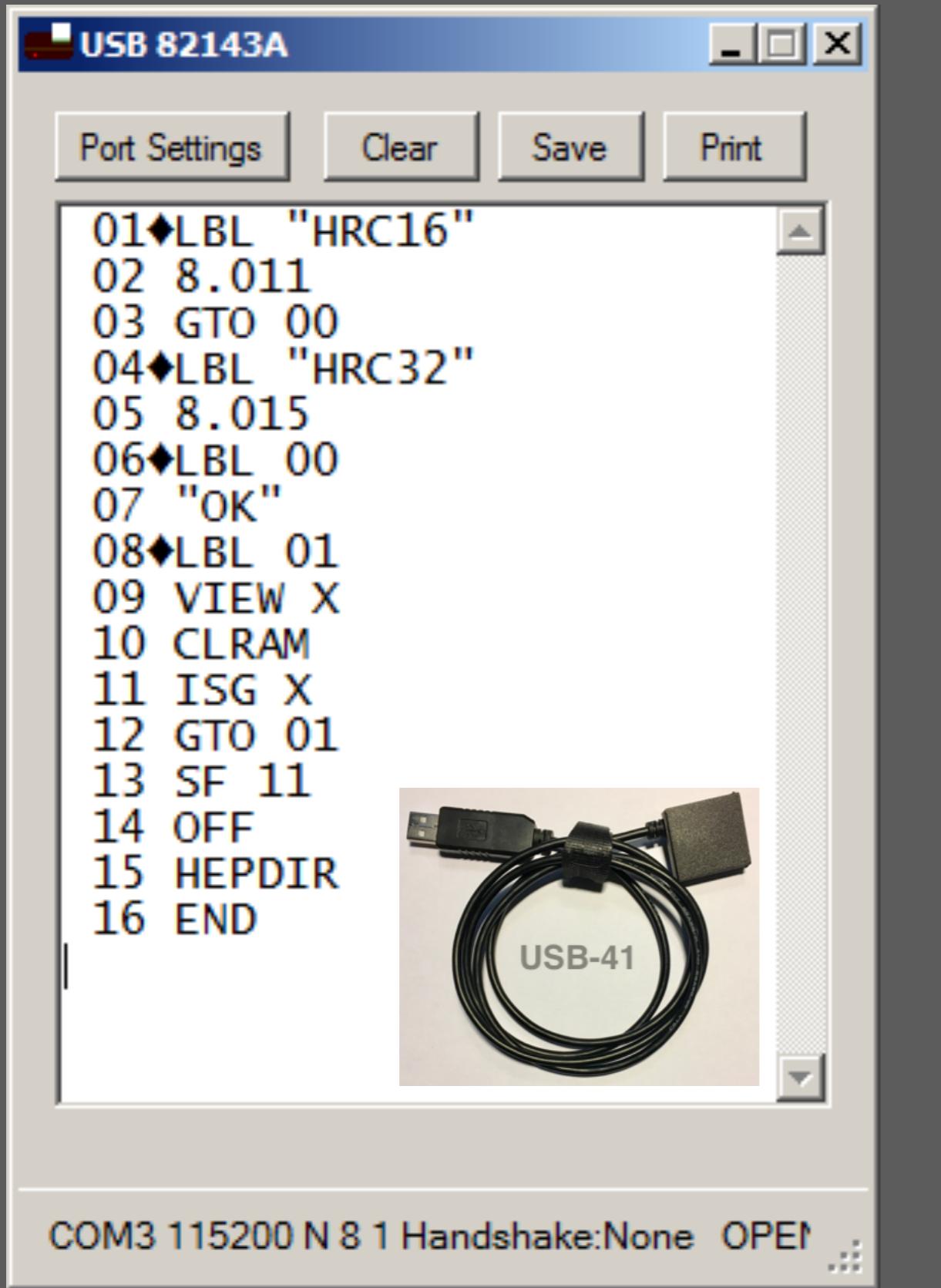
PICkit 2 Programmer

- The purpose of this application is to upload/download a PIC Extended Intel HEX file to/from the microcontroller.
- PICkit 2 is a Windows .NET application developed by Microchip.
- PICkit 2 last release is March 2009
- Available at pickit2.software.informer.com
- Included in Clonix & NoV Configuration Utility v6.1 package.



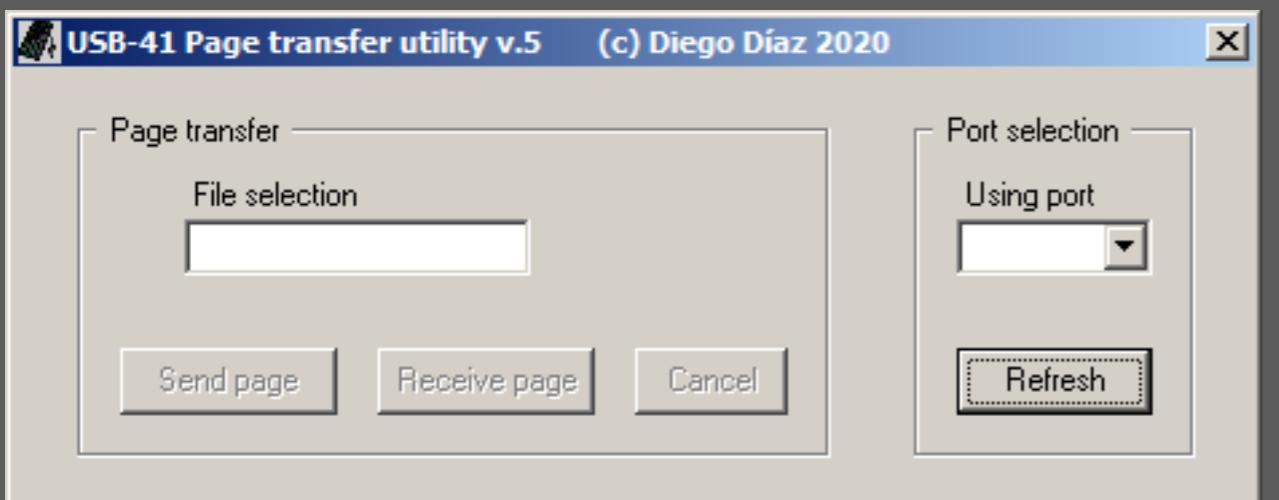
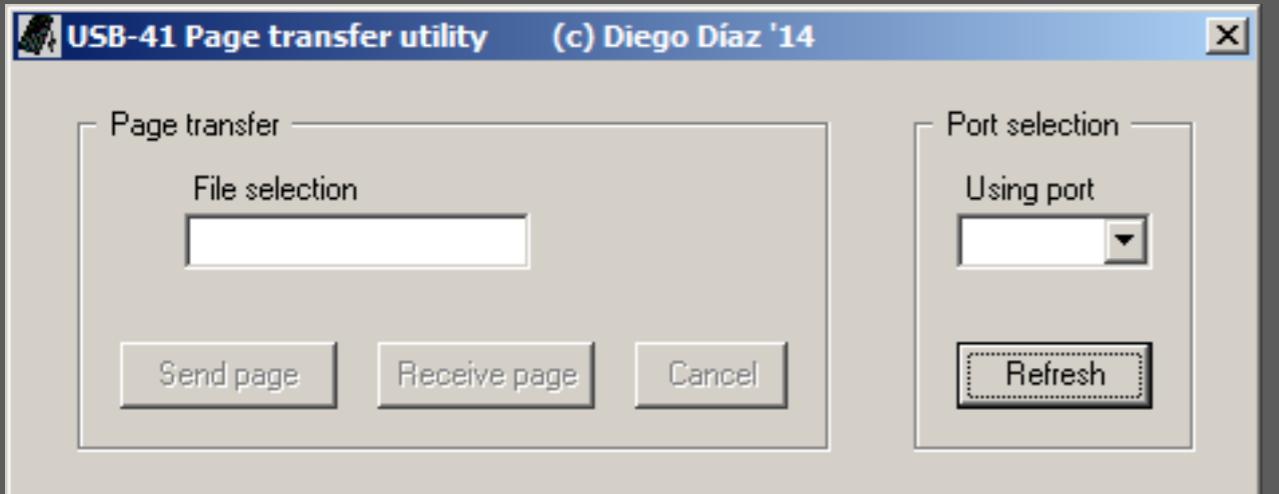
Windows Device Manager

- Windows Device Manager allows you to see on which virtual serial port the USB-41 module has been assigned to.
- Prolific USB-to-Serial Comm Port is used in USB-41 1st Generation.
- FTDI USB-to-Serial Comm Port is used in USB-41 2nd Generation.



USB 82143A

- HP-82143A simulation includes:
 - A USB-82143A Windows application on PC side.
 - A USB-41 module on HP-41C side.
 - Both pieces are needed to fully simulate a HP-82143A printer.
- Text box content can be:
 - Printed.
 - Saved to a RTF file.
 - Copied & pasted into another Windows application.



USB-41 Page Transfer

- This application with a properly configured USB-41 module allows to:
- Directly transfer a 4K ROM from an HP-41 to a ROM image file on a Windows PC.
- Directly transfer a 4K ROM image file from a Windows PC to a 4K QROM page on an HP-41.

CONFIGURATION

Clonix/NoV Utility v4.2

Table of Content

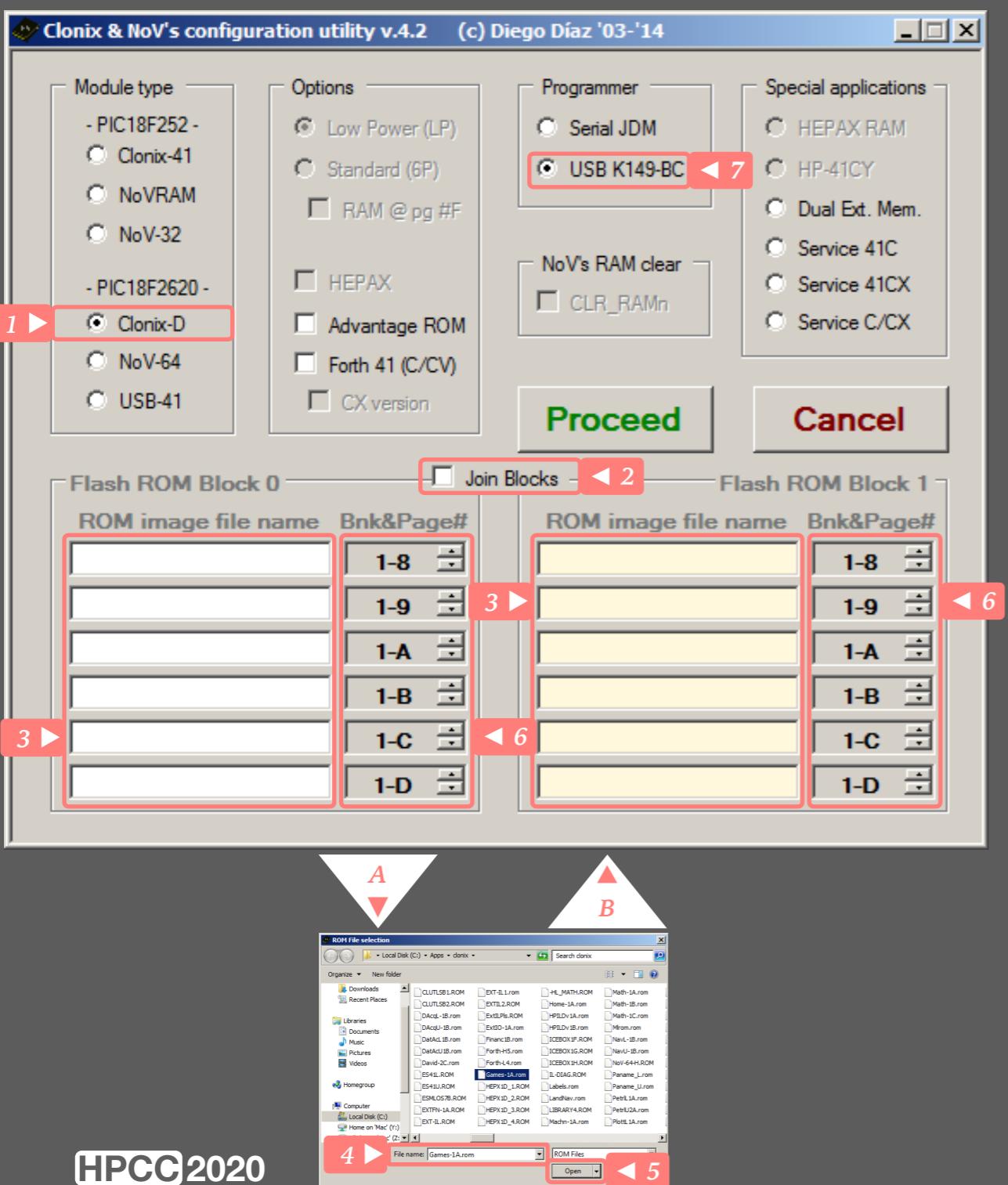
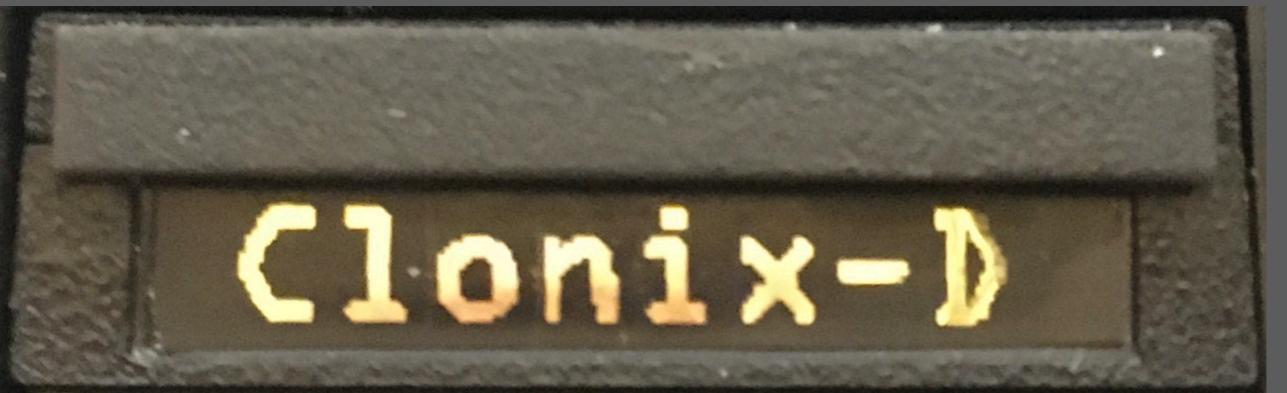
- [Clonix-D](#)
- [NoV-64d](#)
- [USB-41](#)

Clonix-D

Configuration

Table of Content

- ▶ Standard
- ▶ Standard + Merged Blocks
- ▶ Advantage
- ▶ Forth 41 C/CV
- ▶ Forth 41 CX
- ▶ Advantage + Forth 41 C/CV
- ▶ Advantage + Forth 41 CX
- ▶ Dual X-Memory
- ▶ Service 41C/CV
- ▶ Service 41CX
- ▶ Service C/CX



Standard

.....

Goal: loading ROMs into the module.

1. Select Clonix-D option.

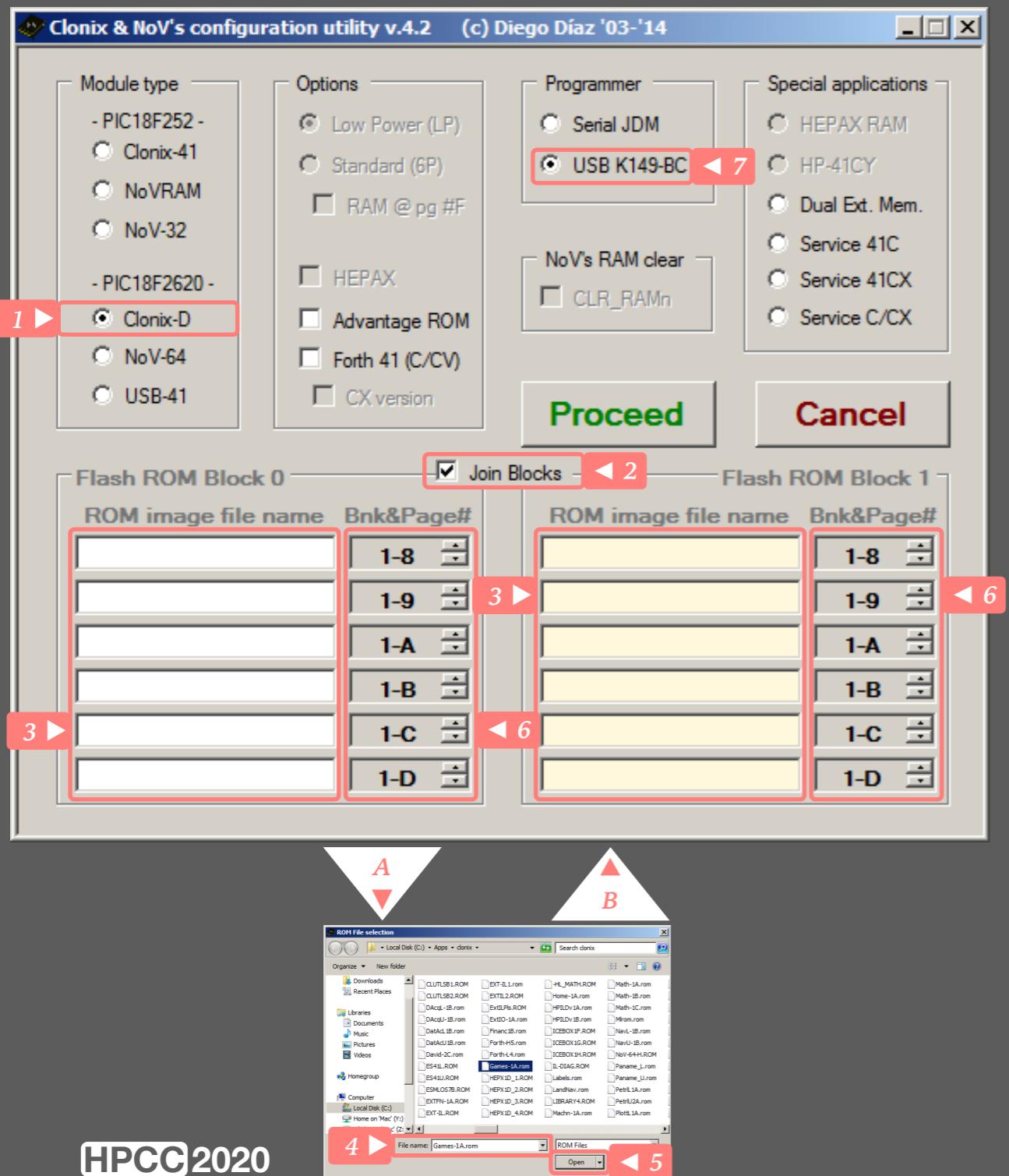
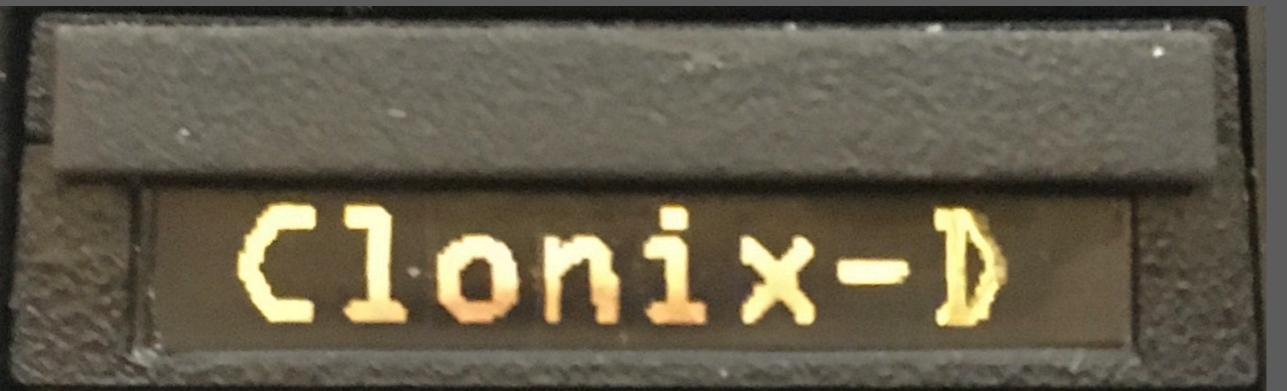
2. Unselect Join Blocks.

Flash ROM Block 0 (white) and Block 1 (yellow) are two separated blocks. ROM's specified in Flash ROM Block 0 (white) will be visible when the module is inserted into an odd port, while Flash ROM Block 1 (yellow) will be visible when the module is inserted into an even port.

For each ROM file you want to map:

3. Click in one the ROM image file name white space to show file selection dialog.
4. Select ROM file name.
5. Click on Open button.
6. Select the Bank [1..4] & Page [#4..#F] you want to map your ROM image to.

Go to Programming section.



Standard + Merged Blocks

.....

Goal: loading ROMs into the module. Flash blocks are merged allowing more ROM's to be mapped.

1. Select Clonix-D option.

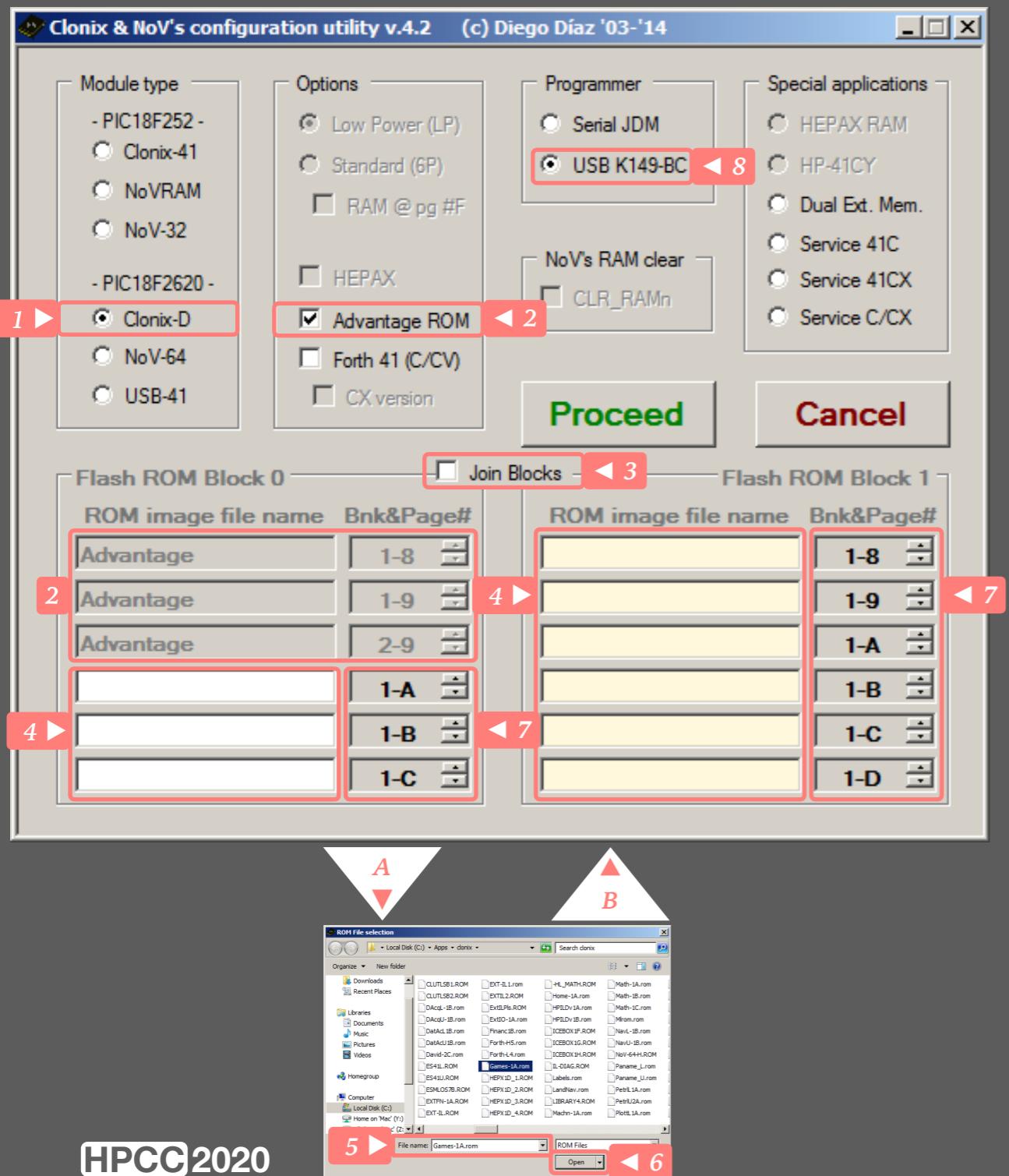
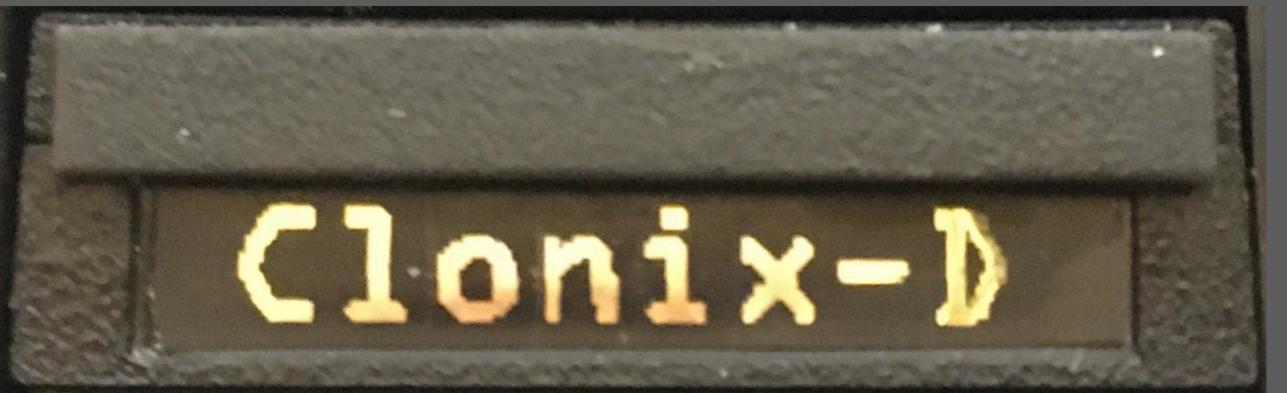
2. Select Join Blocks.

Flash ROM Block 0 (white) and Block 1 (yellow) are merged into a single block.

For each ROM file you want to map:

3. Click in one the ROM image file name white space to show file selection dialog.
4. Select ROM file name.
5. Click on Open button.
6. Select the Bank [1..4] & Page [#4..#F] you want to map you ROM image to.

Go to Programming section.



Advantage

.....

Goal: loading HP Advantage ROM and optionally other ROMs into the module.

1. Select Clonix-D option.

2. Select Advantage ROM.

Load ROM images at pages #8, #9 & #9 bank 2.

3. Optional: unselect or select Join Blocks.

Flash ROM Block 0 (white) and Block 1 (yellow) are either two separated blocks or merged into a single block.

For each ROM file you want to map:

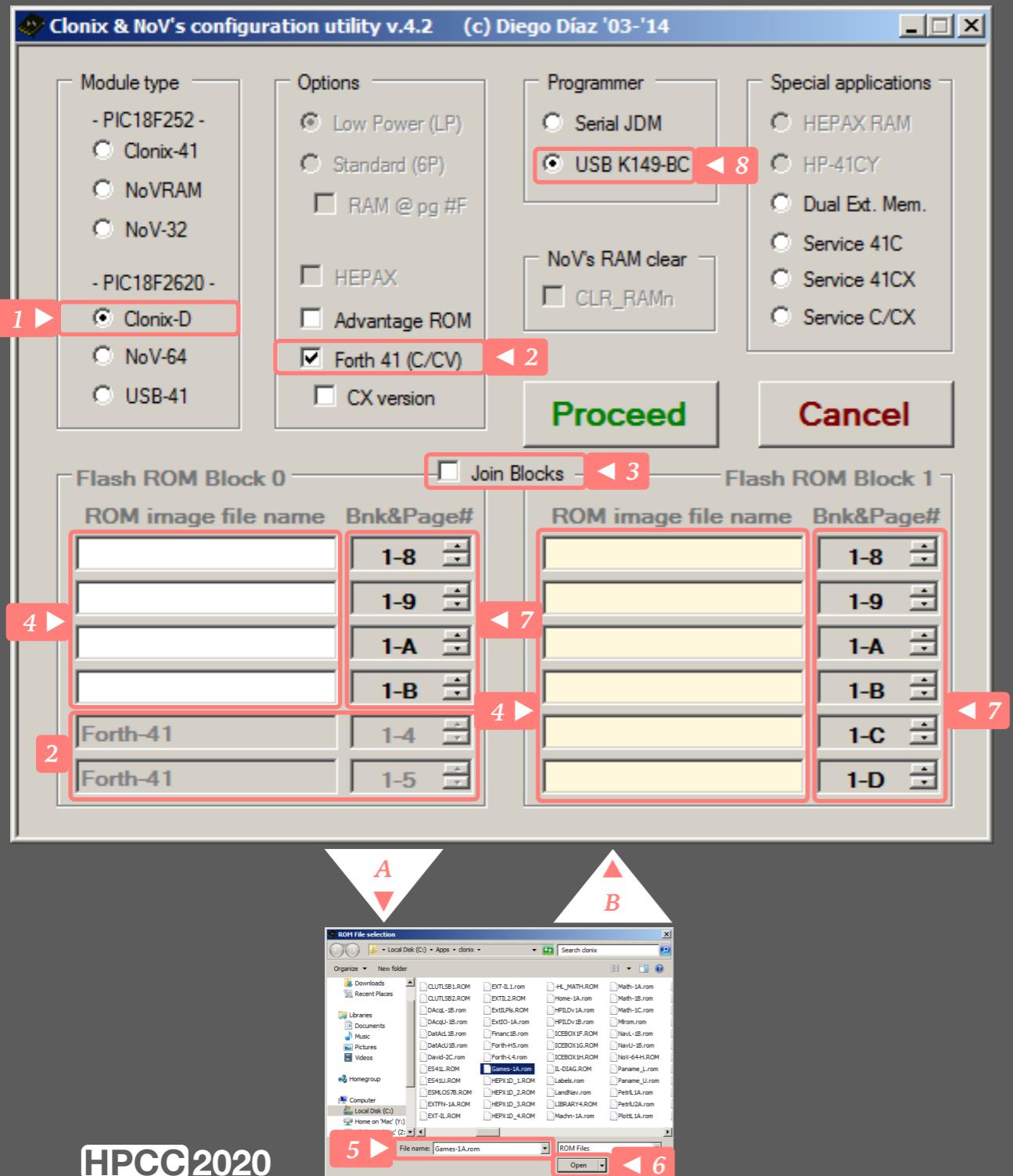
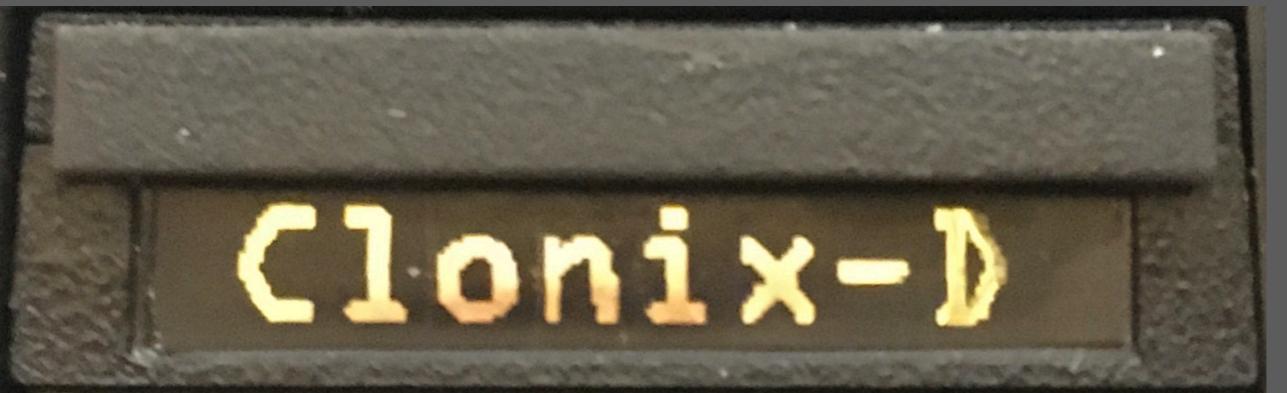
4. Click in one the ROM image file name white space to show file selection dialog.

5. Select ROM file name.

6. Click on Open button.

7. Select the Bank [1..4] & Page [#4..#F] you want to map your ROM image to.

Go to Programming section.



Forth 41 C/CV

.....

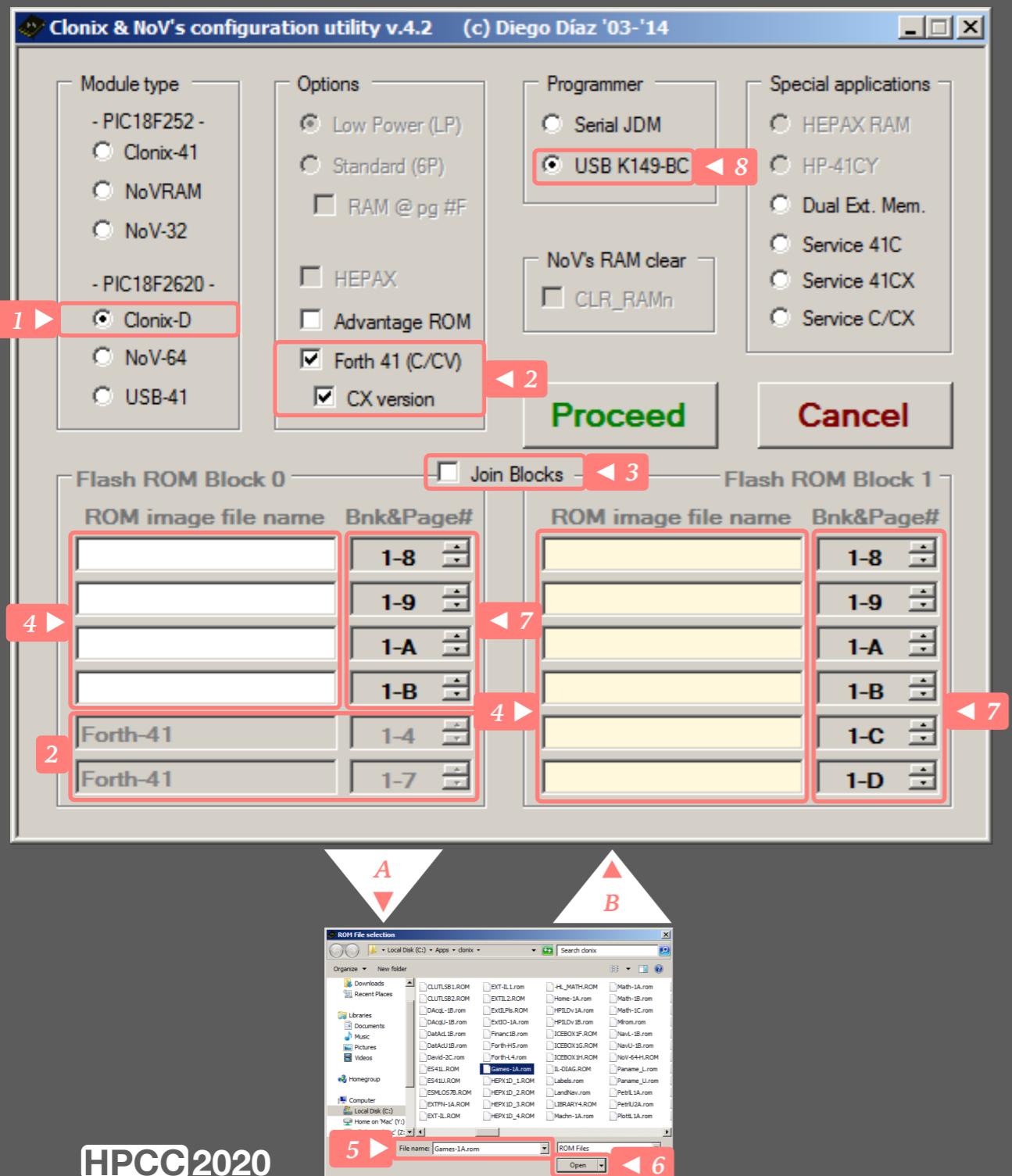
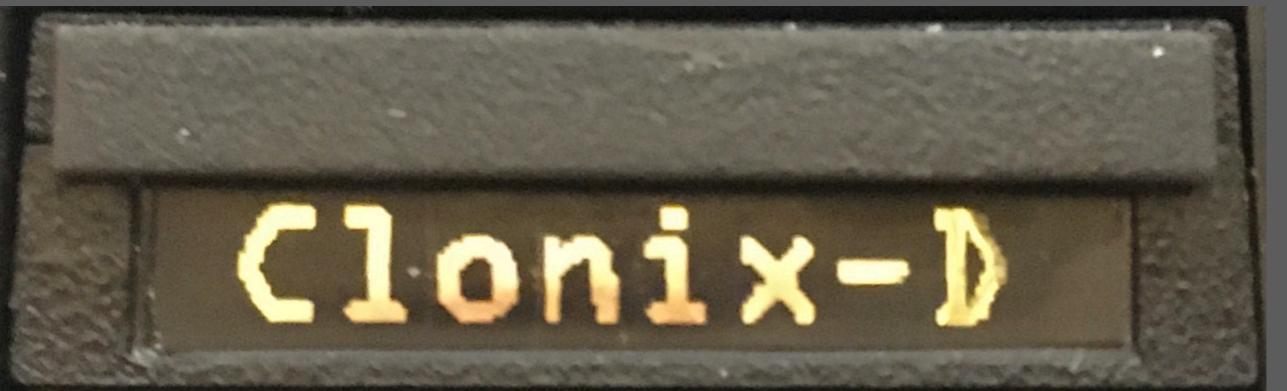
Goal: loading a subset version of the Forth language for the 41C/CV and optionally other ROMs into the module.

1. Select Clonix-D option.
2. Select Forth 41 (C/CV).
Load ROM images at pages #4 & #5.
3. Optional: unselect or select Join Blocks.
Flash ROM Block 0 (white) and Block 1 (yellow) are either two separated blocks or merged into a single block.

For each ROM file you want to map:

4. Click in one the ROM image file name white space to show file selection dialog.
5. Select ROM file name.
6. Click on Open button.
7. Select the Bank [1..4] & Page [#4..#F] you want to map your ROM image to.

Go to Programming section.



Forth 41 CX

.....

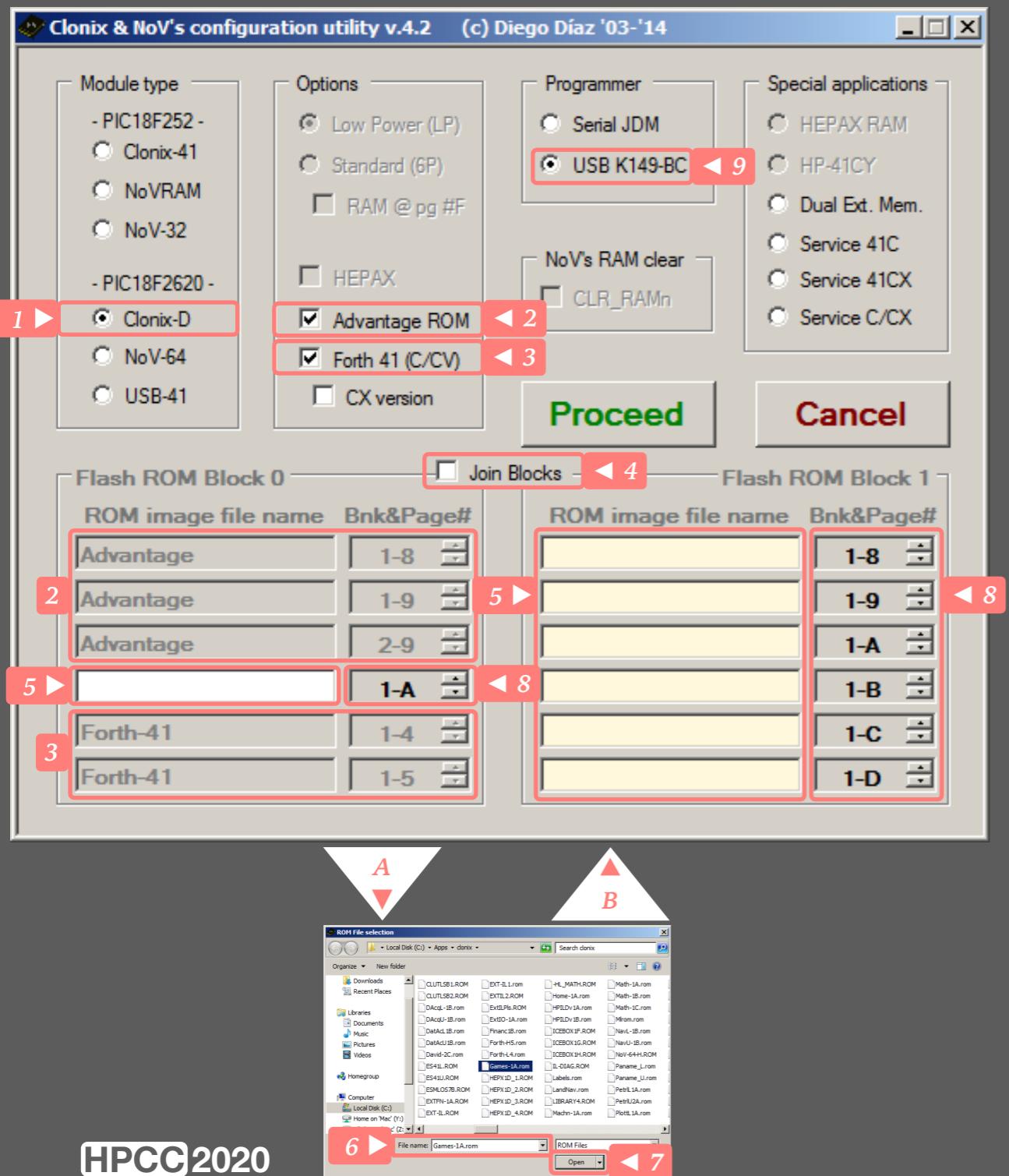
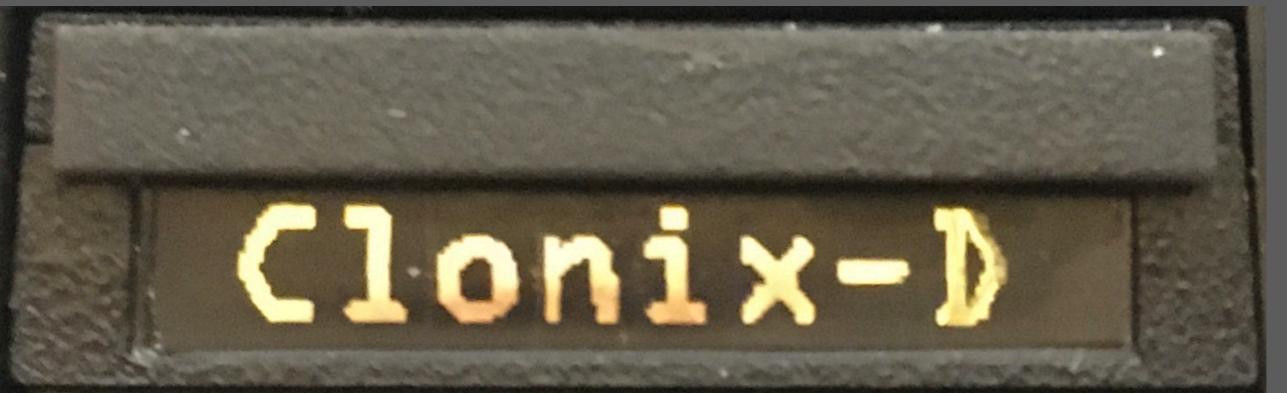
Goal: loading a subset version of the Forth language for the 41CX and optionally other ROMs into the module.

1. Select Clonix-D option.
2. Select Forth 41 (C/CV).
Load ROM images at pages #4 & #7.
3. Optional: unselect or select Join Blocks.
Flash ROM Block 0 (white) and Block 1 (yellow) are either two separated blocks or merged into a single block.

For each ROM file you want to map:

4. Click in one the ROM image file name white space to show file selection dialog.
5. Select ROM file name.
6. Click on Open button.
7. Select the Bank [1..4] & Page [#4..#F] you want to map you ROM image to.

Go to Programming section.



Advantage + Forth 41 C/CV

Goal: loading HP Advantage ROM, a subset version of the Forth language for the 41C/CV and optionally other ROMs into the module.

1. Select Clonix-D option.

2. Select Advantage ROM.

Load ROM images at pages #8, #9 & #9 bank 2.

3. Select Forth 41 (C/CV).

Load ROM images at pages #4 & #5.

4. Optional: unselect or select Join Blocks.

Flash ROM Block 0 (white) and Block 1 (yellow) are either two separated blocks or merged into a single block.

For each ROM file you want to map:

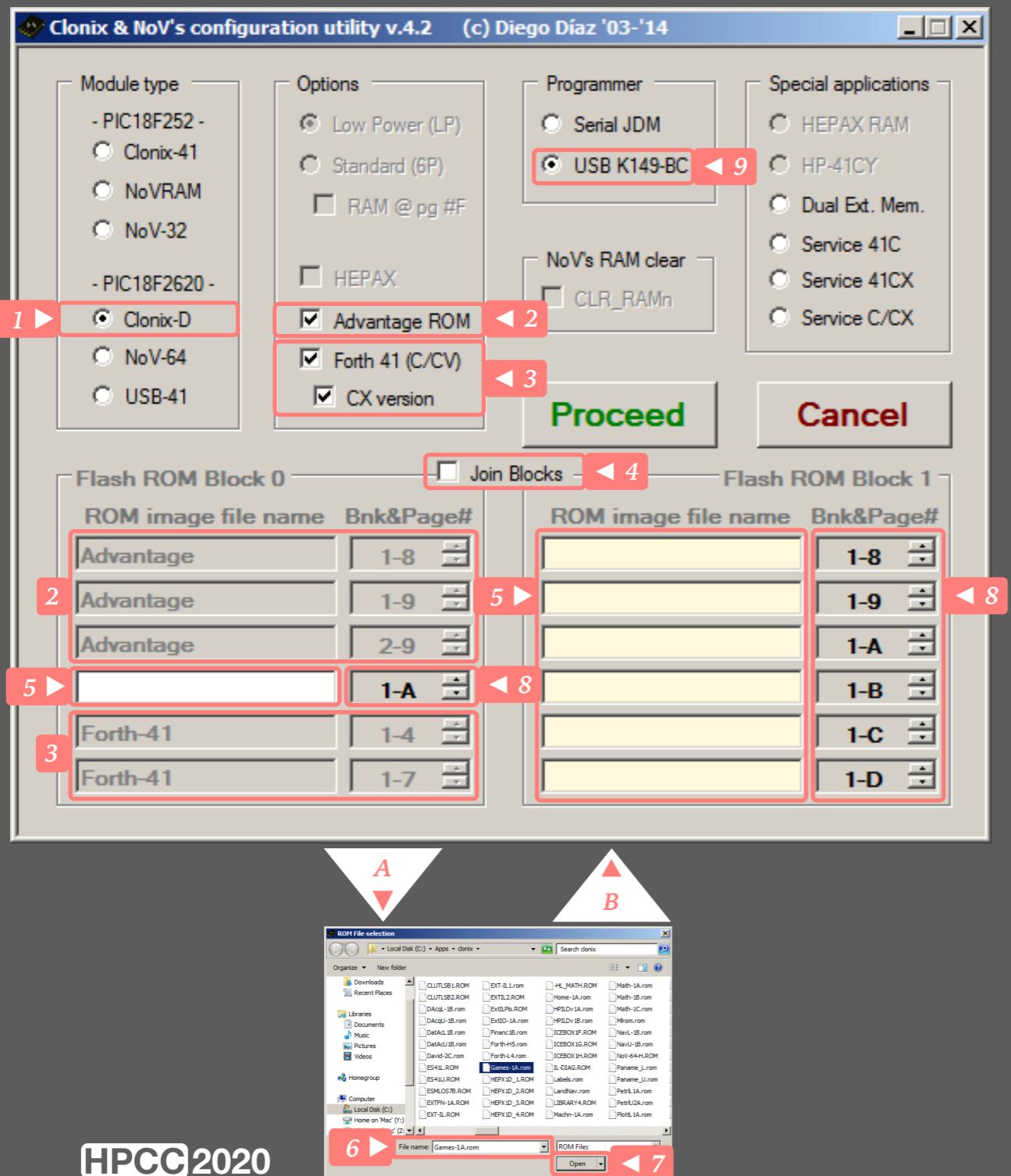
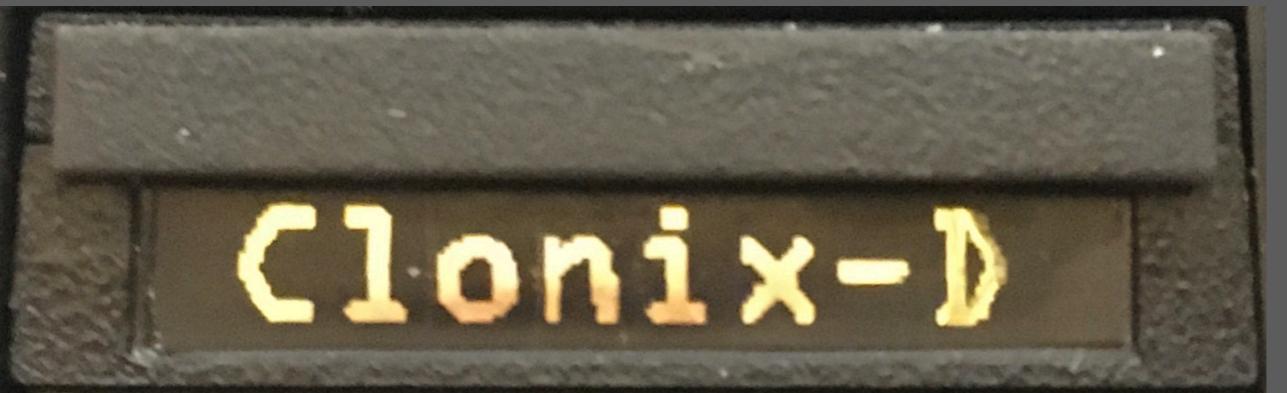
5. Click in one the ROM image file name white space to show file selection dialog.

6. Select ROM file name.

7. Click on Open button.

8. Select the Bank [1..4] & Page [#4..#F] you want to map your ROM image to.

Go to Programming section.



Advantage + Forth 41 CX

.....

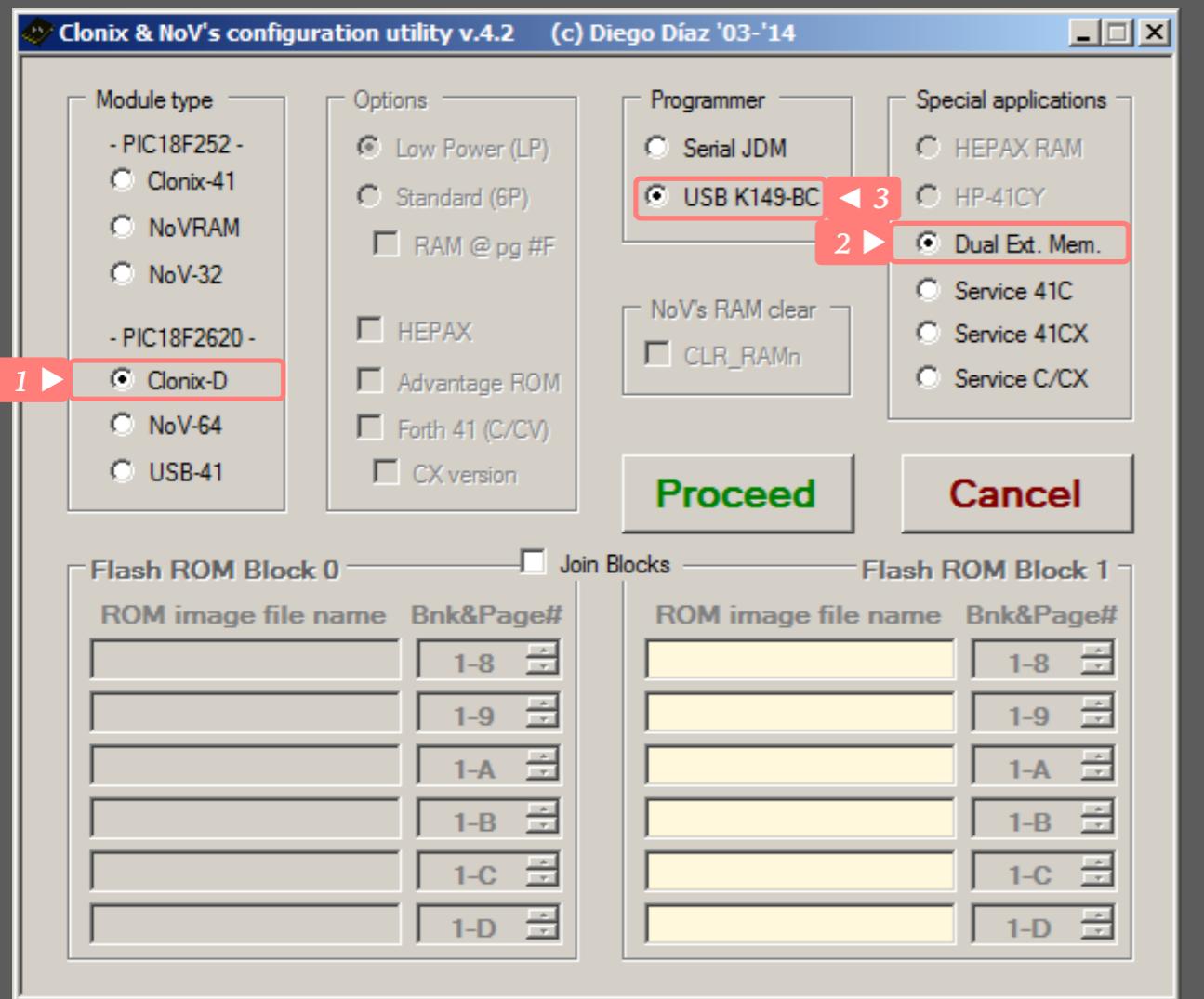
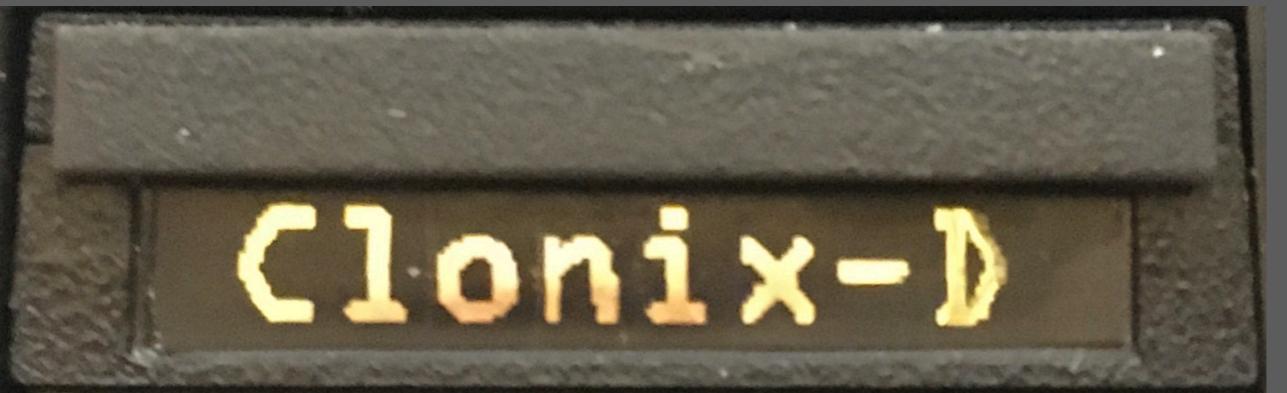
Goal: loading HP Advantage ROM, a subset version of the Forth language for the 41CX and optionally other ROMs into the module.

1. Select Clonix-D option.
2. Select Advantage ROM.
Load ROM images at pages #8, #9 & #9 bank 2.
3. Select Forth 41 (C/CV) then CX version.
Load ROM images at pages #4 & #7.
4. Optional: unselect or select Join Blocks.
Flash ROM Block 0 (white) and Block 1 (yellow) are either two separated blocks or merged into a single block.

For each ROM file you want to map:

5. Click in one the ROM image file name white space to show file selection dialog.
6. Select ROM file name.
7. Click on Open button.
8. Select the Bank [1..4] & Page [#4..#F] you want to map your ROM image to.

Go to Programming section.



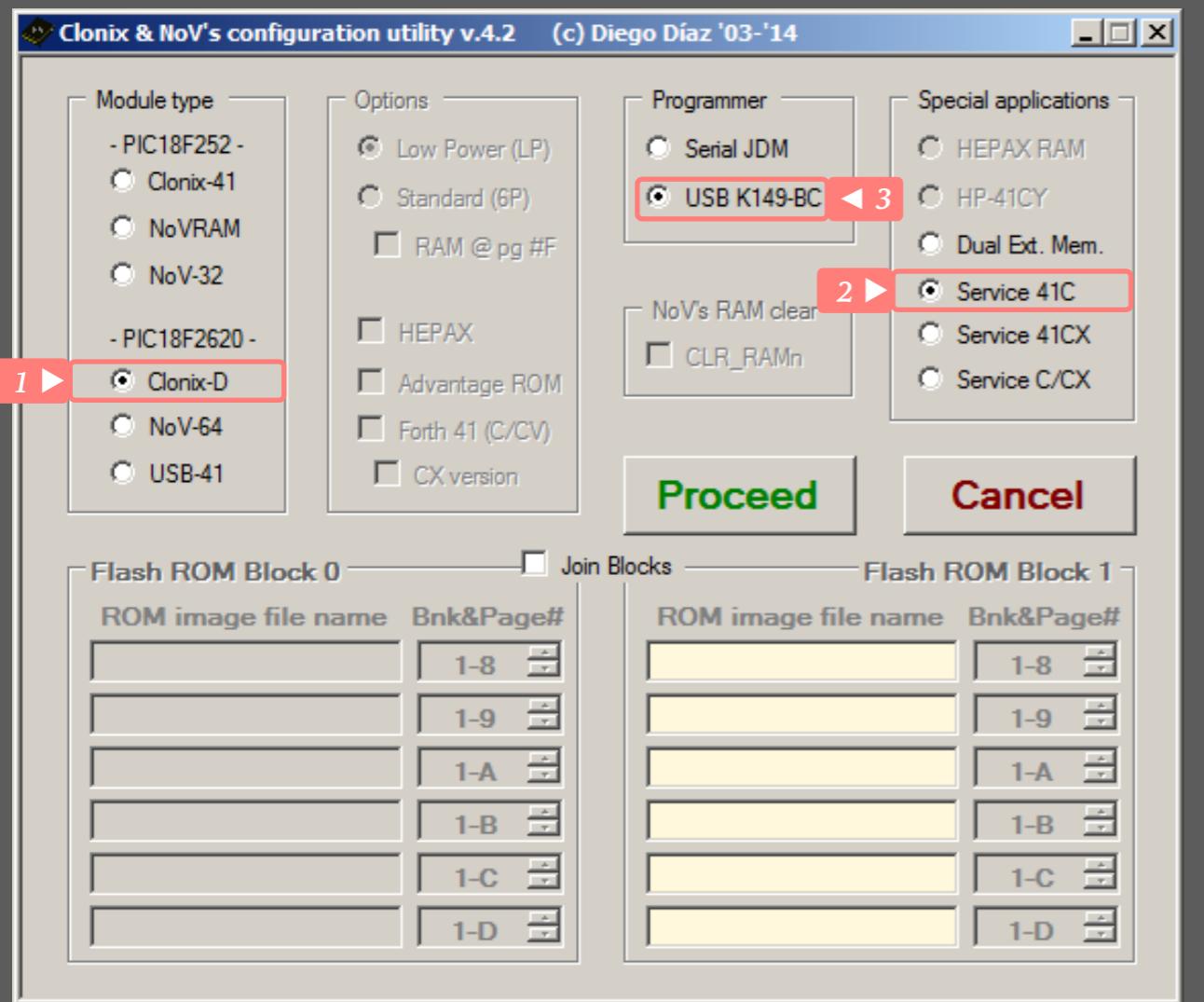
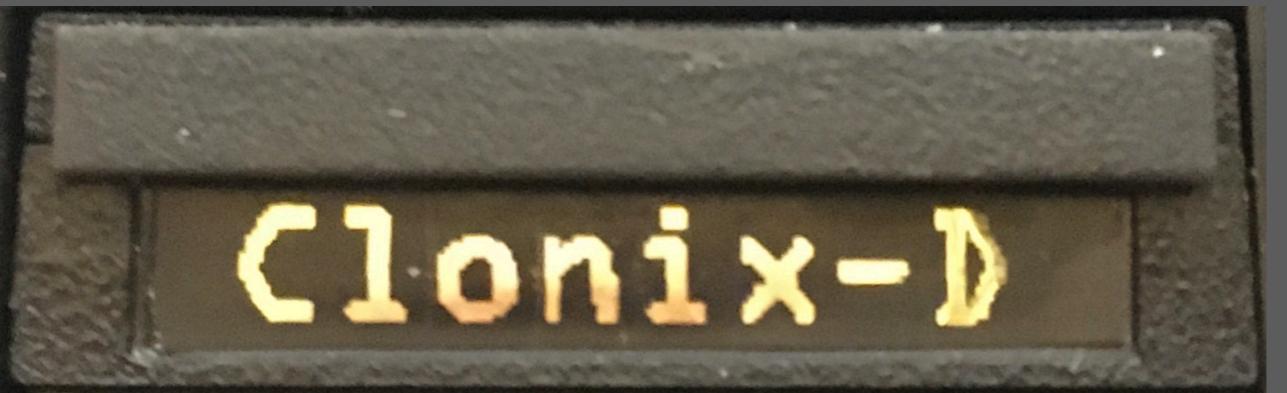
Dual X-Memory

Goal: loading a program into the module that simulate two 82181A X-Memory modules.

1. Select Clonix-D option.
2. Select Dual Ext. Mem. to configure the module as a Double X-Memory module.

This configuration add 476 of Extended-Registers RAM to the system. RAM content is lost when the module is unplugged from the calculator.

Go to Programming section.



Service 41C/CV

.....

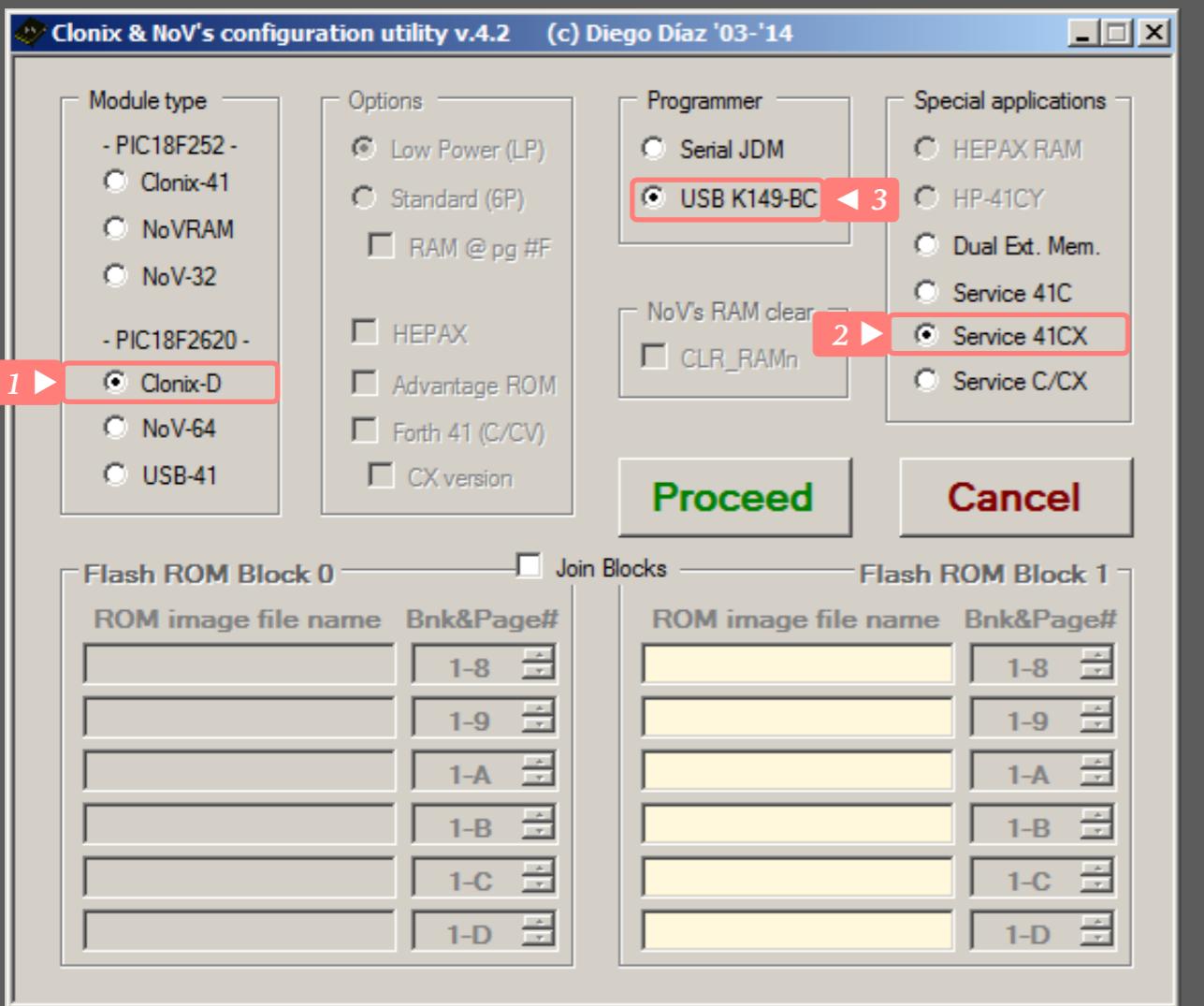
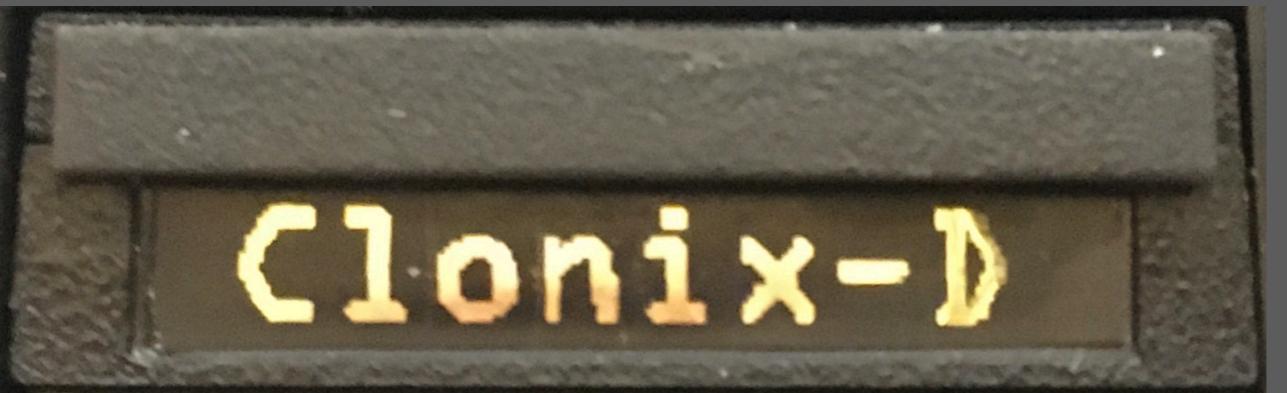
Goal: loading HP Service ROM into the module. Support: 41C, 41CV, RAM (x1 & x4), ROM (4K & 8K) & Card Reader.

1. Select Clonix-D option.

2. Select Service 41C.

Load HP Service ROM [SM-1C] image in page #4.

Go to Programming section.



Service 41CX

.....

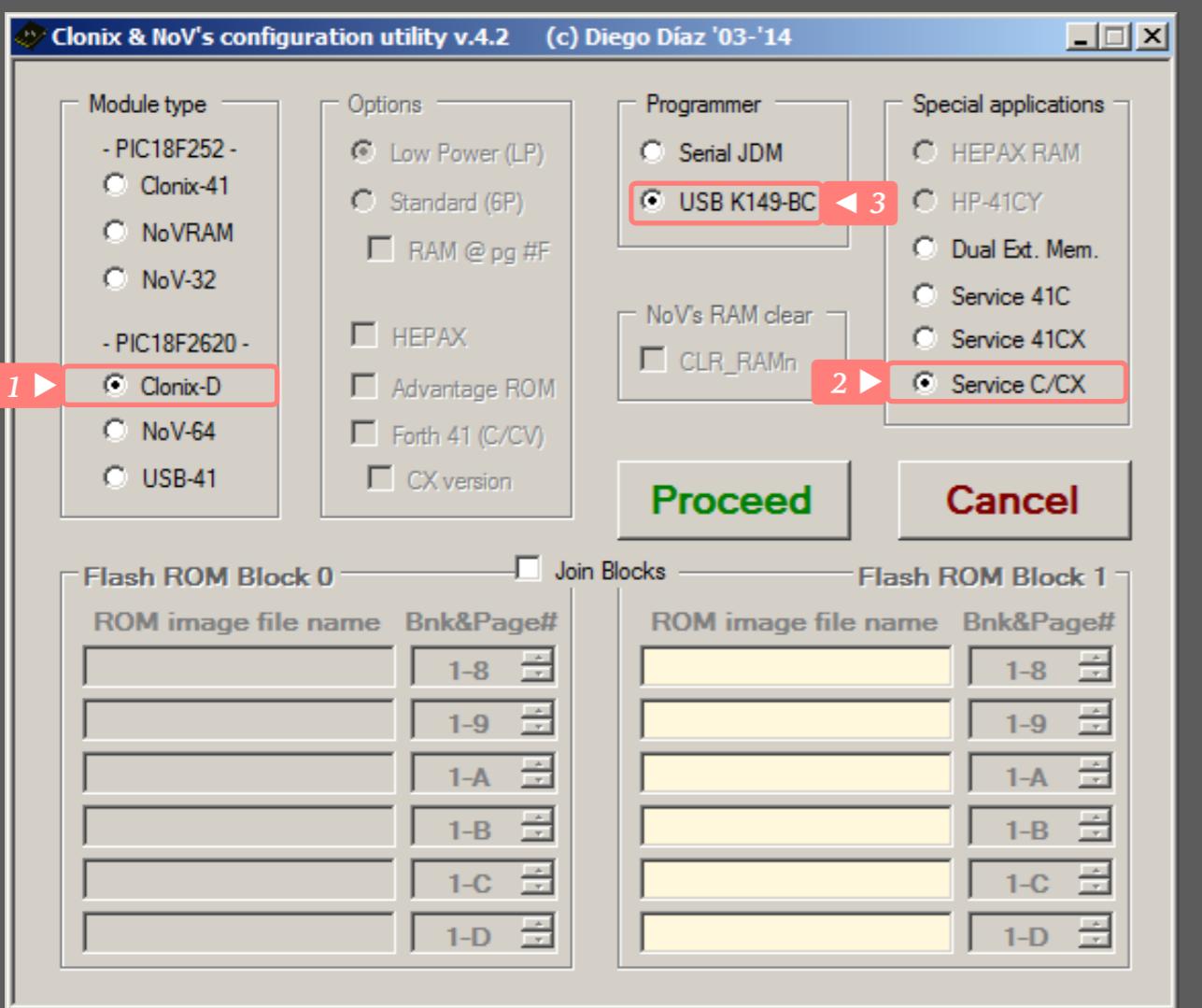
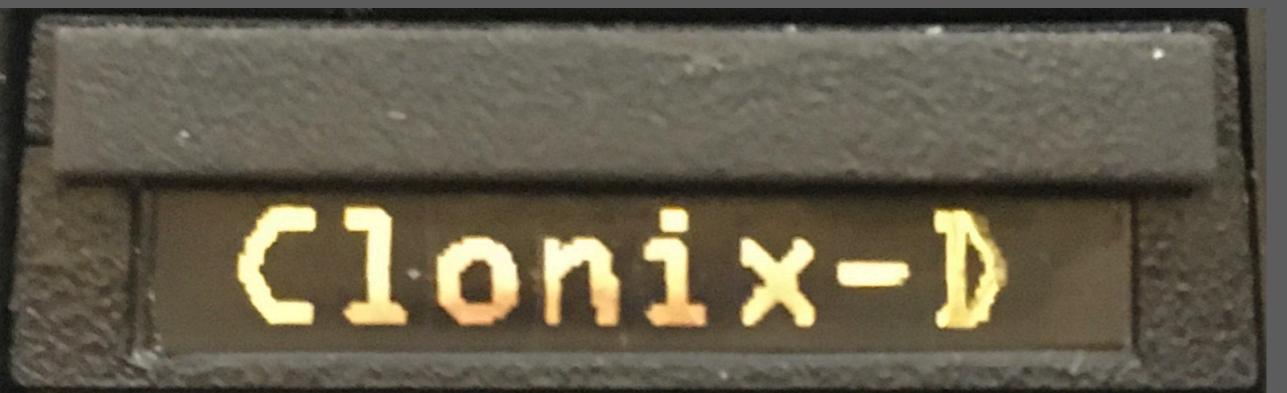
Goal: loading HP Service ROM into the module. Support: 41CV, 41CX, Time, X-Fnc, X-Mem, RAM (x1 & x4), ROM (4K to 16K).

1. Select Clonix-D option.

2. Select Service 41CX.

Load HP Service ROM [SM-2A] image in page #4.

Go to Programming section.



Service C/CX

.....

Goal: loading HP Service ROMs into the module.

Note: this option was first created for the Clonix 41d Anniversary Ed.

1. Select Clonix-D option.

2. Select Service C/CX.

Load Service 41C/CV ROM [SM-1C] image in page #4 block 0.

Load Service 41CX ROM [SM-2A] image in page #4 block 1.

Inserting the module into an odd port activates block 0.

Inserting the module into an even port activates block 1.

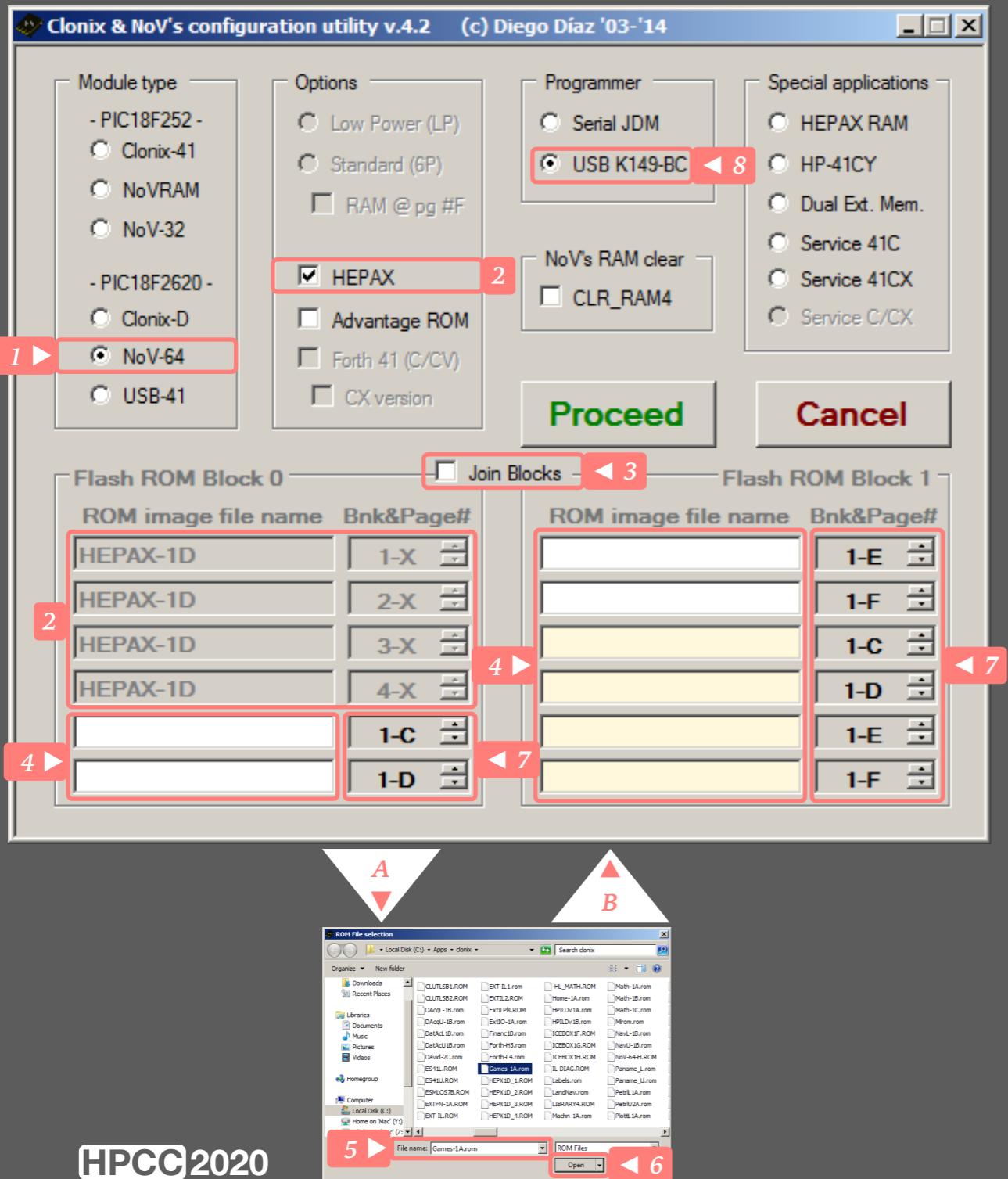
Go to Programming section.

NoV-64(d)

Configuration

Table of Content

- HEPAX
- HEPAX + Merged Blocks
- HEPAX + Advantage
- NoV's RAM Clear
- HEPAX RAM
- HP-41CY & RAMBOX64
- Dual X-Memory
- Service 41C/CV
- Service 41CX
- Quasi-ROM
- Clonix-D Persona



HEPAX

.....

Goal: loading Advanced HEPAX ROM and optionally other ROMs into the module.

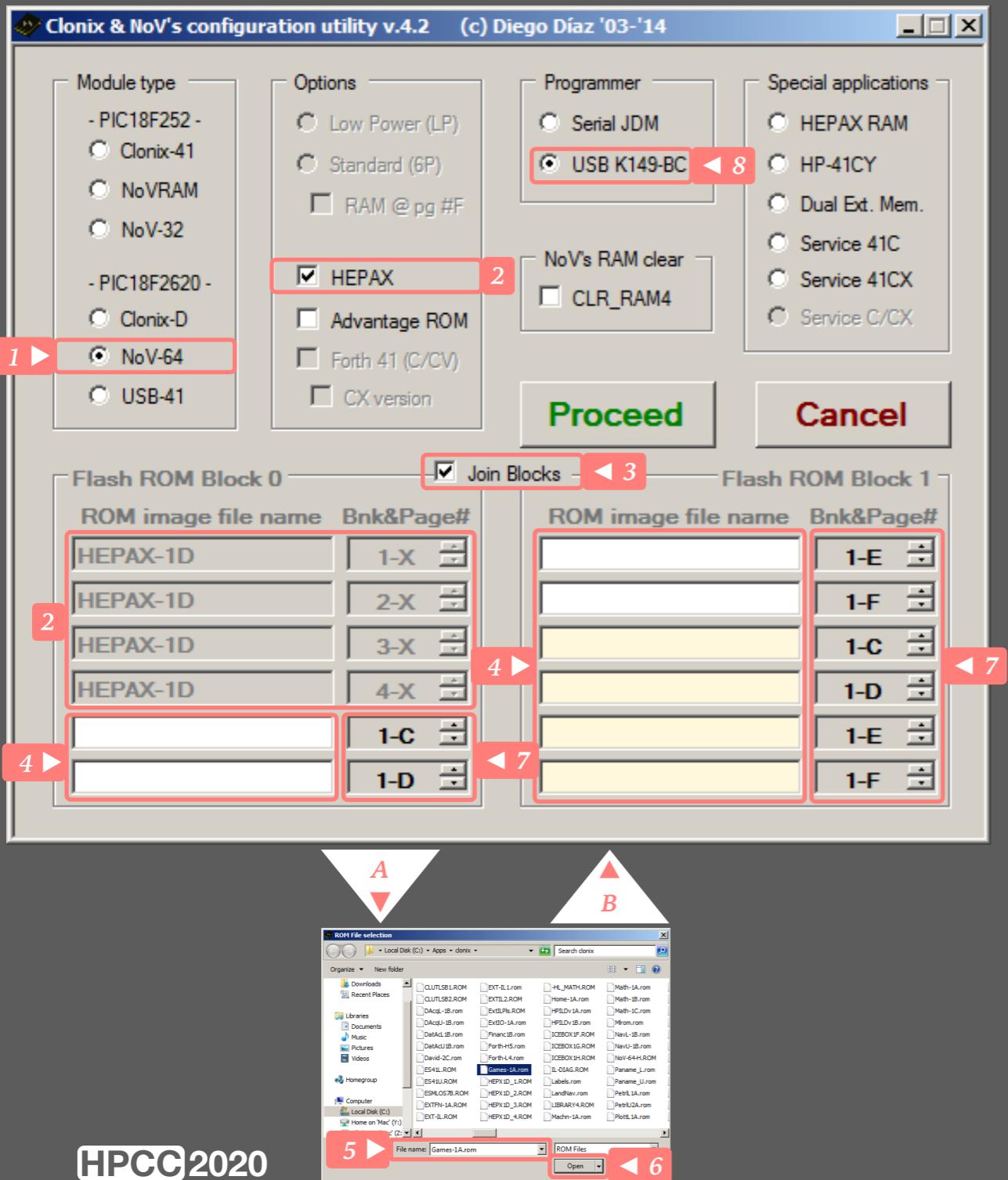
1. Select **NoV-64** option.
2. **HEPAX** is automatically selected.
3. Unselect **Join Blocks**.

Flash ROM Block 0 (white) and Block 1 (yellow) are two separated blocks.

For each ROM file you want to map:

4. Click in one the **ROM image file name** white space to show file selection dialog.
5. Select ROM file name.
6. Click on **Open** button.
7. Select the **Bank [1..4]** & **Page [#4..#F]** you want to map your ROM image to.

Go to Programming section.



HEPAX + Merged Blocks

.....

Goal: loading Advanced HEPAX ROM and optionally other ROMs into the module. Flash blocks are merged allowing more ROM's to be mapped.

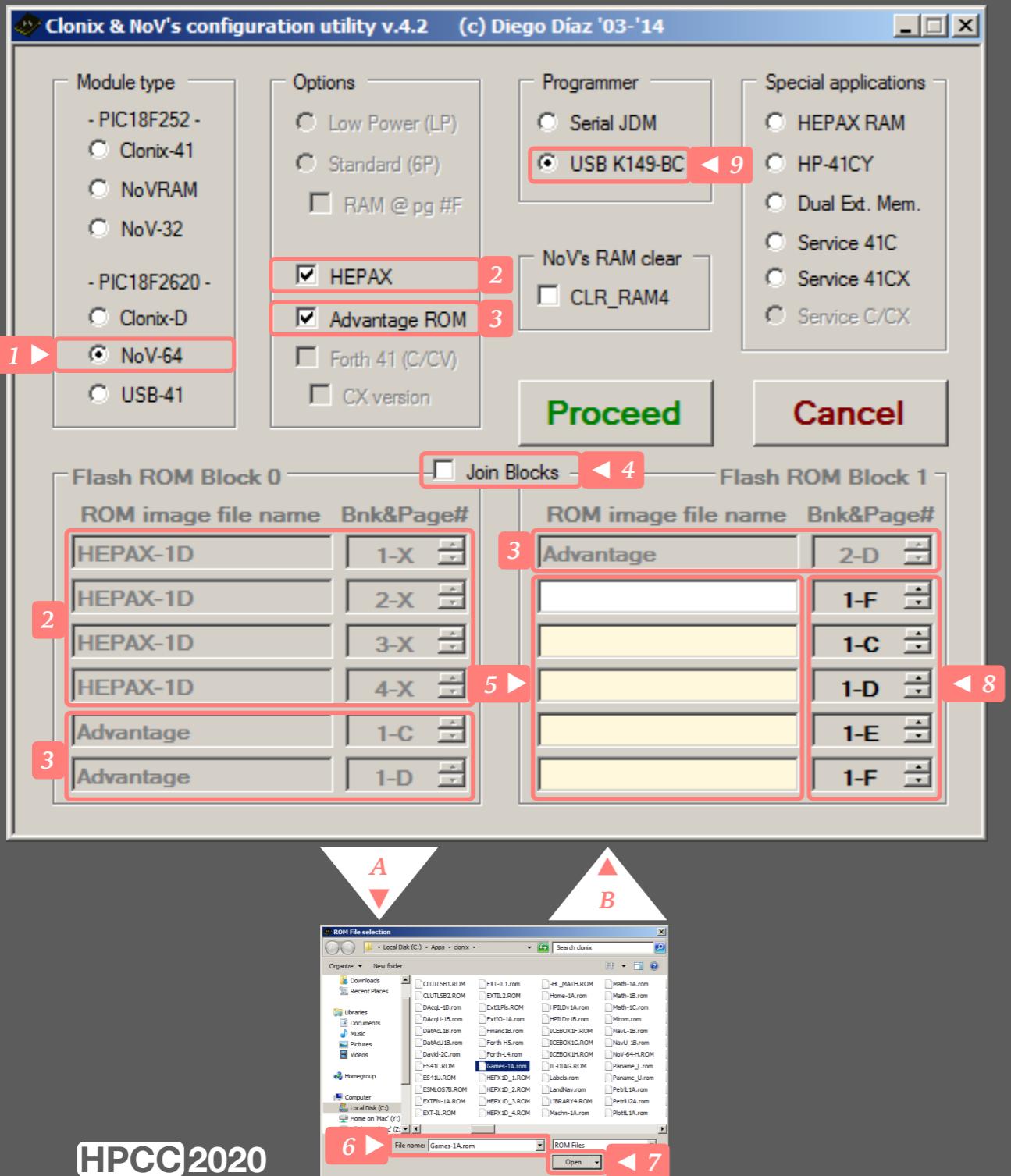
1. Select NoV-64 option.
2. HEPAX is automatically selected.
3. Select Join Blocks.

Flash ROM Block 0 (white) and Block 1 (yellow) are merged into a single block.

For each ROM file you want to map:

4. Click in one the ROM image file name white space to show file selection dialog.
5. Select ROM file name.
6. Click on Open button.
7. Select the Bank [1..4] & Page [#4..#F] you want to map your ROM image to.

Go to Programming section.



HEPAX + Advantage

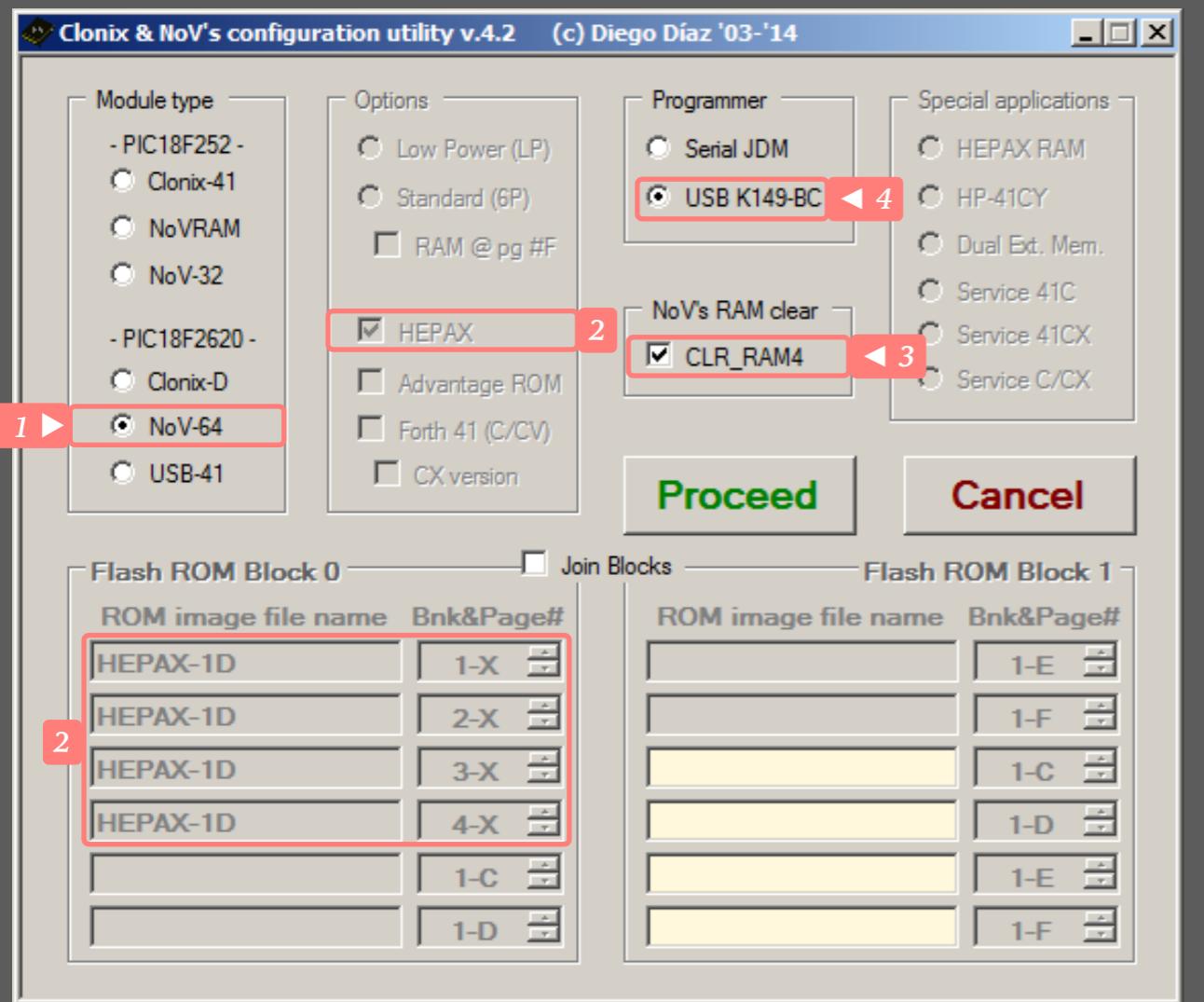
Goal: loading Advanced HEPAX ROM, HP Advantage ROM and optionally other ROMs into the module.

1. Select **NoV-64** option.
2. **HEPAX** is automatically selected.
3. Select **Advantage ROM**.
Load ROM images at pages #C, #D & #D bank 2.
4. Optional: unselect or select **Join Blocks**.
Flash ROM Block 0 (white) and Block 1 (yellow) are either two separated blocks or merged into a single block.

For each ROM file you want to map:

5. Click in one the **ROM image file name** white space to show file selection dialog.
6. Select ROM file name.
7. Click on **Open** button.
8. Select the **Bank [1..4]** & **Page [#4..#F]** you want to map your ROM image to.

Go to Programming section.

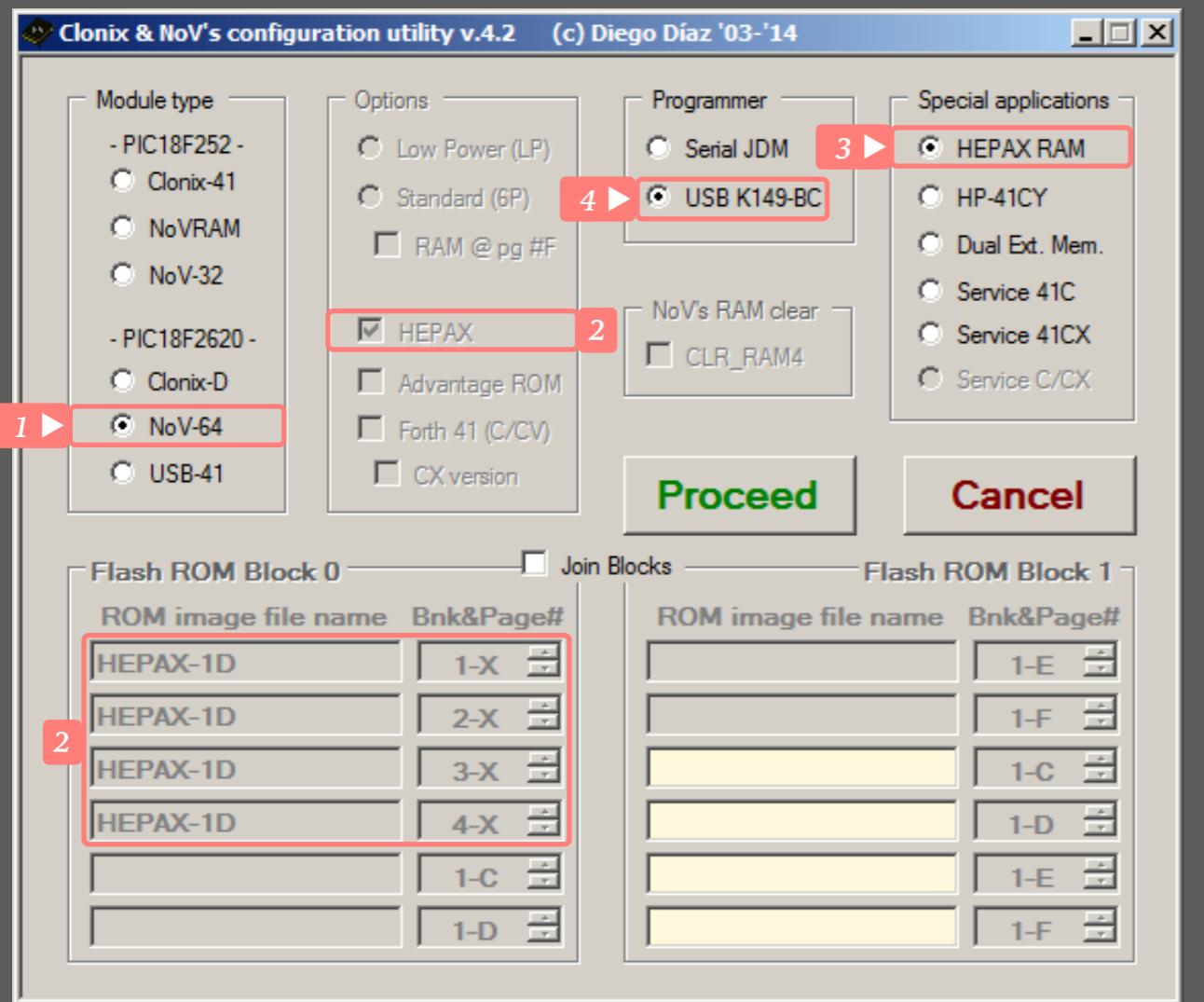


NoV's RAM Clear

Goal: loading a program into the module that clears the NoV module RAM.

1. Select **NoV-64** option.
2. **HEPAX** is automatically selected but unused.
3. Select **CLR_RAM4** to load a specialized firmware that will clear HEPAX RAM.
Note: this option has been proven to be unreliable, more details in [Clearing HEPAX RAM](#) section.

Go to Programming section.



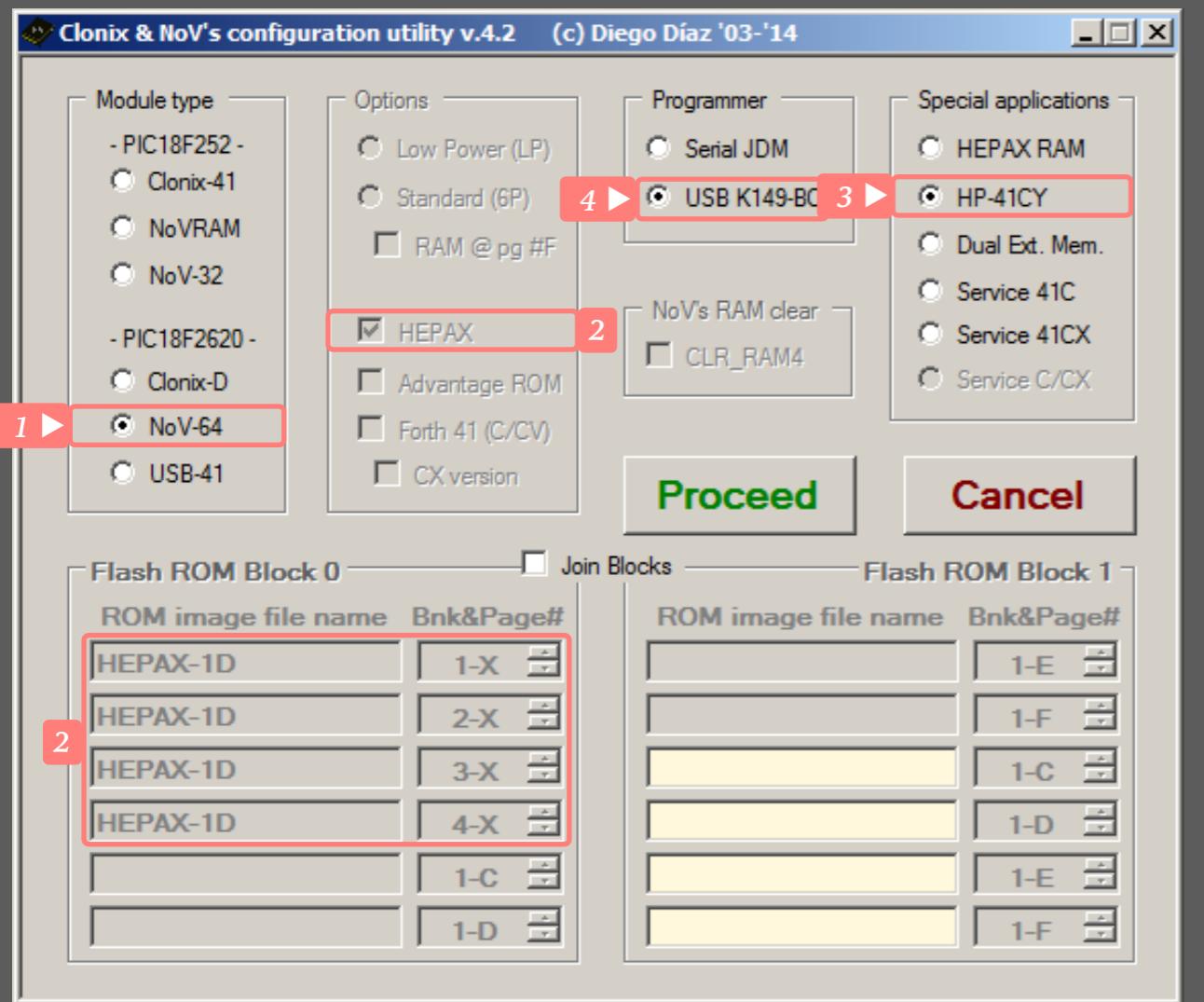
HEPAX RAM

.....

Goal: loading a program into the module that simulate an HEPAX Double Memory module.

1. Select **NoV-64** option.
2. **HEPAX** is automatically selected but unused.
3. Select **HEPAX RAM** to configure the module as a HEPAX Double Memory unit.

Go to Programming section.



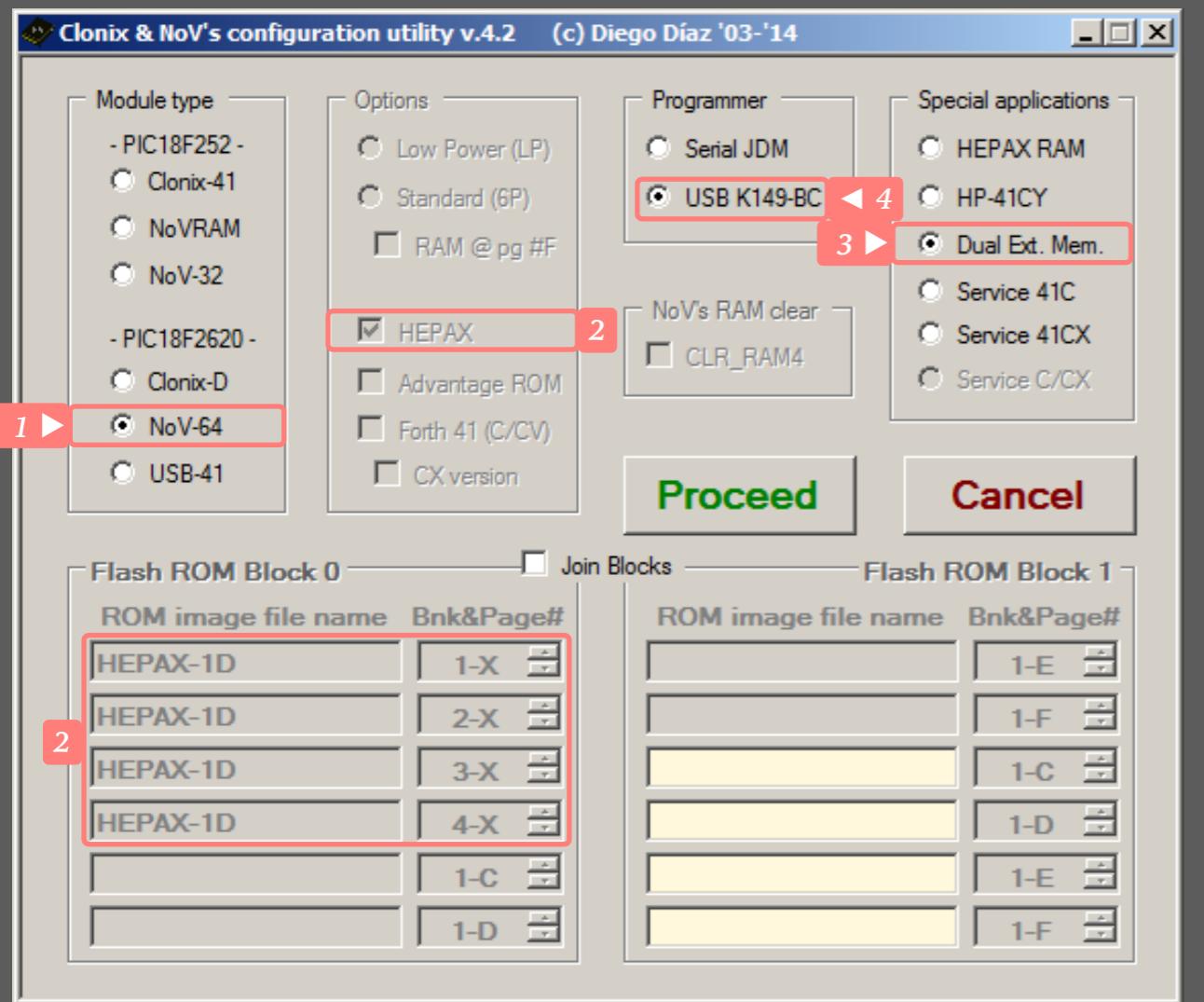
HP-41CY & RAMBOX64

Goal: loading W&W RAMBOX64 ROM into the module to simulate a RAMBOX64 unit or an HP-41CY calculator.

1. Select NoV-64 option.
2. HEPAX is automatically selected but unused.
3. Select HP-41CY to configure the module as a W&W RAMBox64 unit.

When inserted into a HP-41CX halfnut you get an HP-41CY replica without the turbo mode.

Go to Programming section.



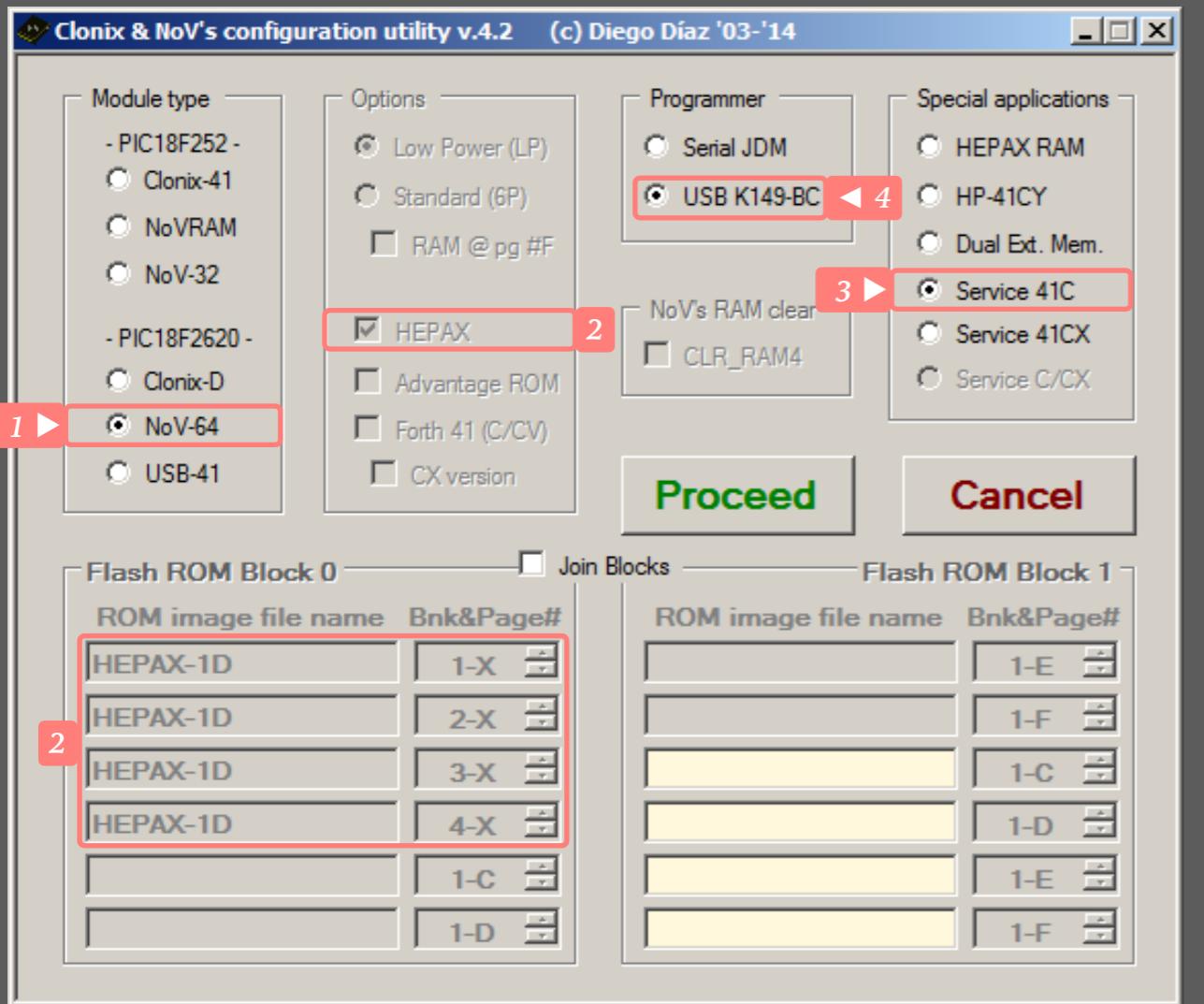
Dual X-Memory

Goal: loading a program into the module that simulate two 82181A X-Memory modules.

1. Select **NoV-64** option.
2. **HEPAX** is automatically selected but unused.
3. Select **Dual Ext. Mem.** to configure the module as a Double X-Memory module.

This configuration add 476 of Extended-Registers RAM to the system. RAM content is lost when the module is unplugged from the calculator.

Go to Programming section.



Service 41C/CV

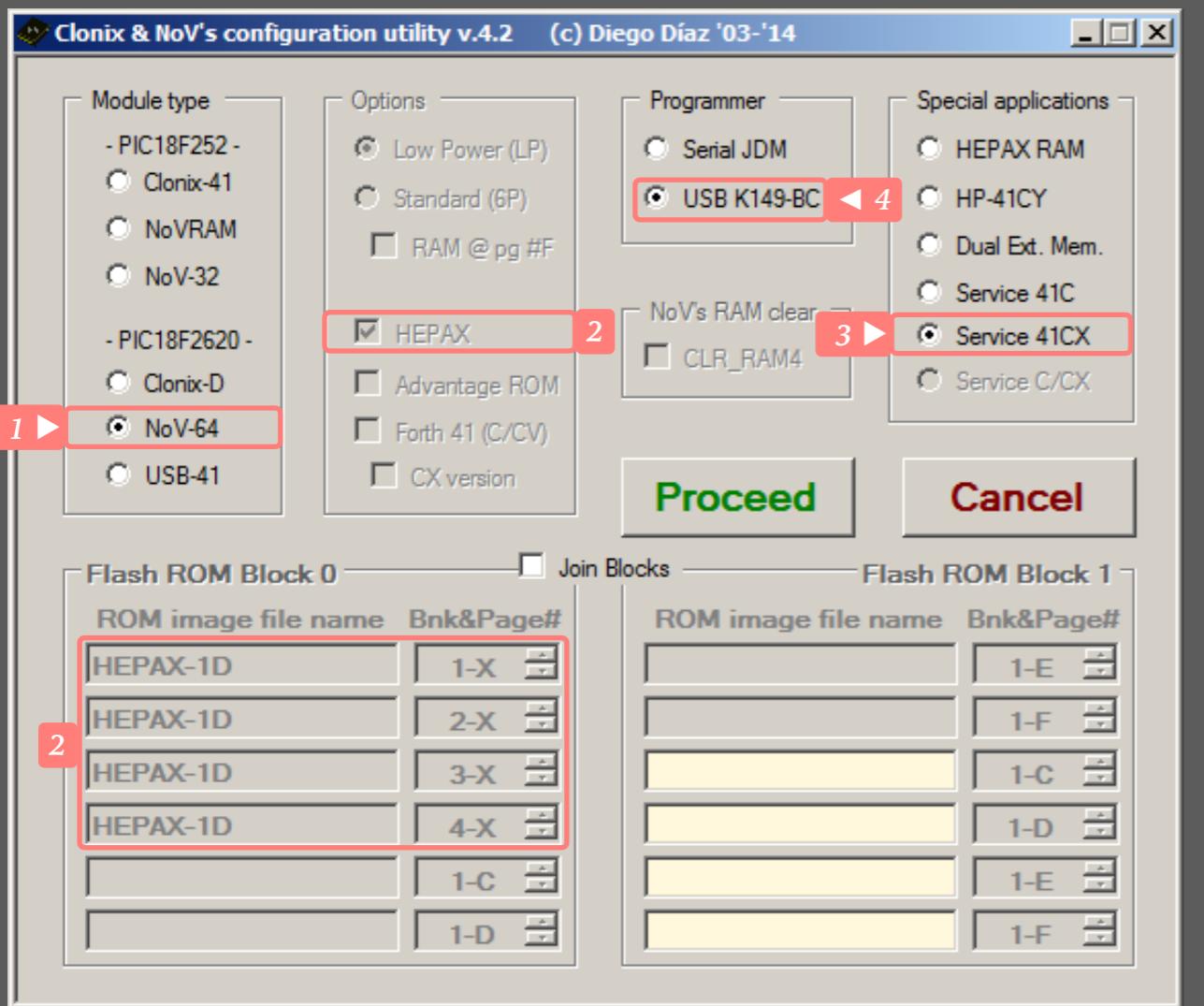
.....

Goal: loading HP Service ROM into the module. Support: 41C, 41CV, RAM (x1 & x4), ROM (4K & 8K) & Card Reader.

1. Select **NoV-64** option.
2. **HEPAX** is automatically selected but unused.
3. Select **Service 41C**.

Load HP Service ROM [SM-1C] image in page #4.

Go to Programming section.



Service 41CX

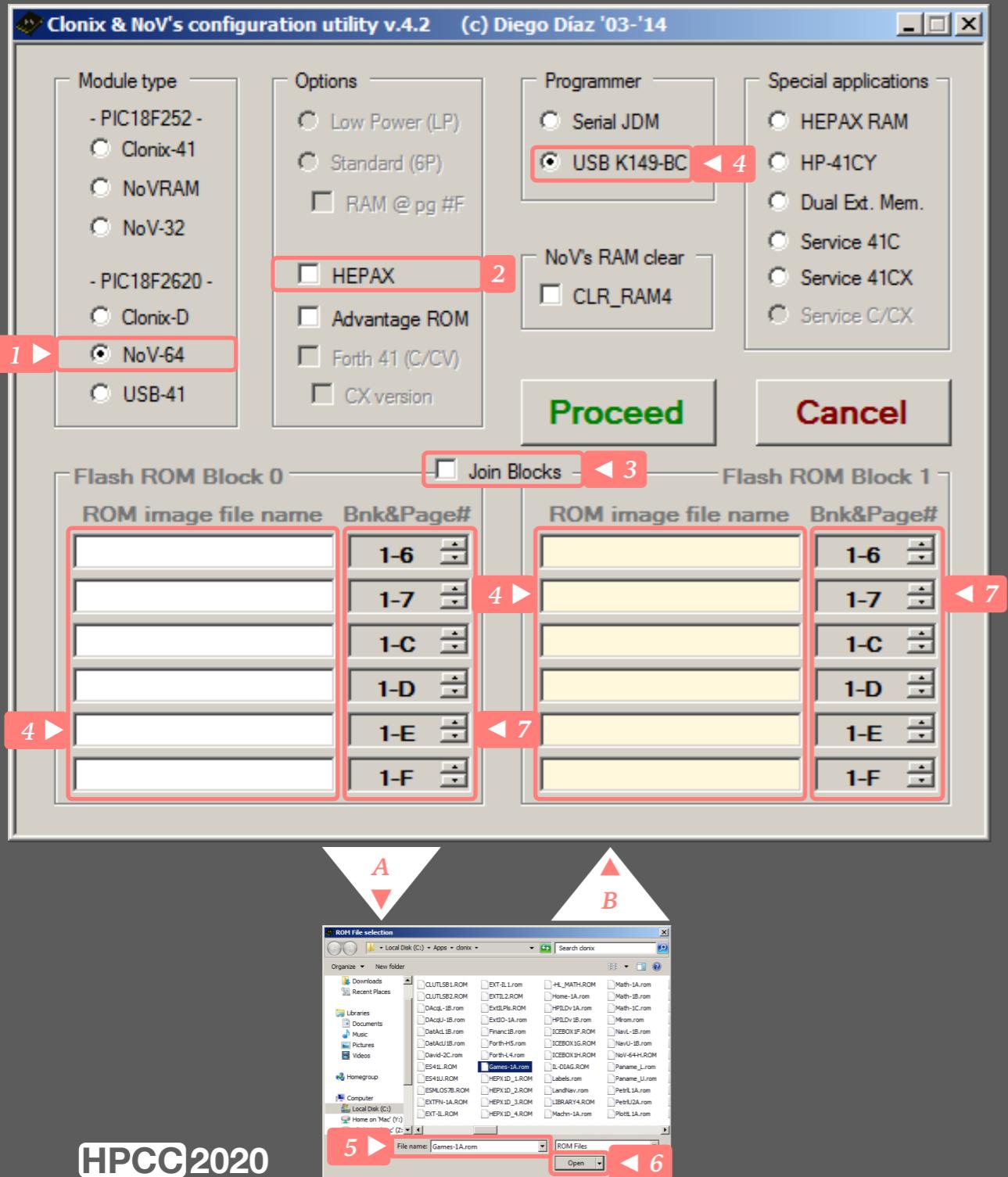
.....

Goal: loading HP Service ROM into the module. Support: 41CV, 41CX, Time, X-Fnc, X-Mem, RAM (x1 & x4), ROM (4K to 16K).

1. Select **NoV-64** option.
2. **HEPAX** is automatically selected but unused.
3. Select **Service 41CX**.

Load HP Service ROM [SM-2A] image in page #4.

Go to Programming section.



Quasi-ROM

.....

Goal: activating RAM/QROM pages #8 to #B and optionally loading ROMs into the module.

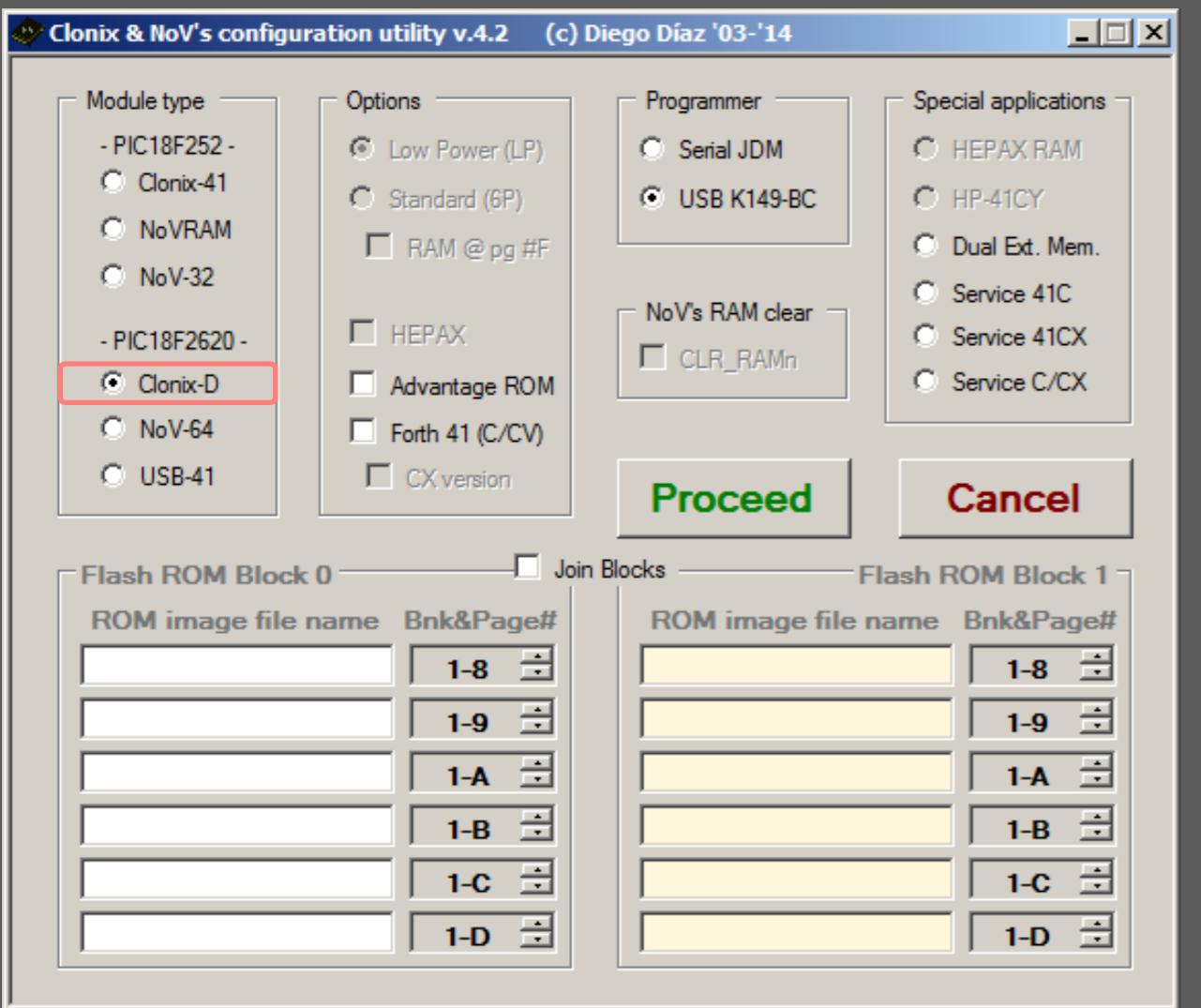
1. Select **NoV-64** option.
2. Unselect **HEPAX** option.
3. Optional: unselect or select **Join Blocks**.

Flash ROM Block 0 (white) and Block 1 (yellow) are either two separated blocks or merged into a single block.

For each ROM file you want to map:

4. Click in one the **ROM image file name** white space to show file selection dialog.
5. Select ROM file name.
6. Click on **Open** button.
7. Select the **Bank [1..4]** & **Page [#4..#F]** you want to map your ROM image to.

Go to Programming section.



Clonix-D Persona

NoV-64d module has the ability to behave exactly like a Clonix-D. If you want that persona, go to the **Clonix-D** section and configure the module without any restrictions.

NoV-64 module has the ability to behave partially like a Clonix-D. If you want that persona, go to the **Clonix-D** section and configure the module with these restrictions:

Join Blocks Unselected

- You can specify a maximum of 6 pages (24K) in Flash ROM Block 1
- Flash ROM Block 0 must contains the same ROM images at the same place as specified in Flash ROM Block 1. (Flash ROM Block 0 is a clone of Flash ROM Block 1)

Join Blocks Selected

- You can specify a maximum of 12 pages (48K) in Flash ROM Block 0+1

USB-41

Configuration

Table of Content

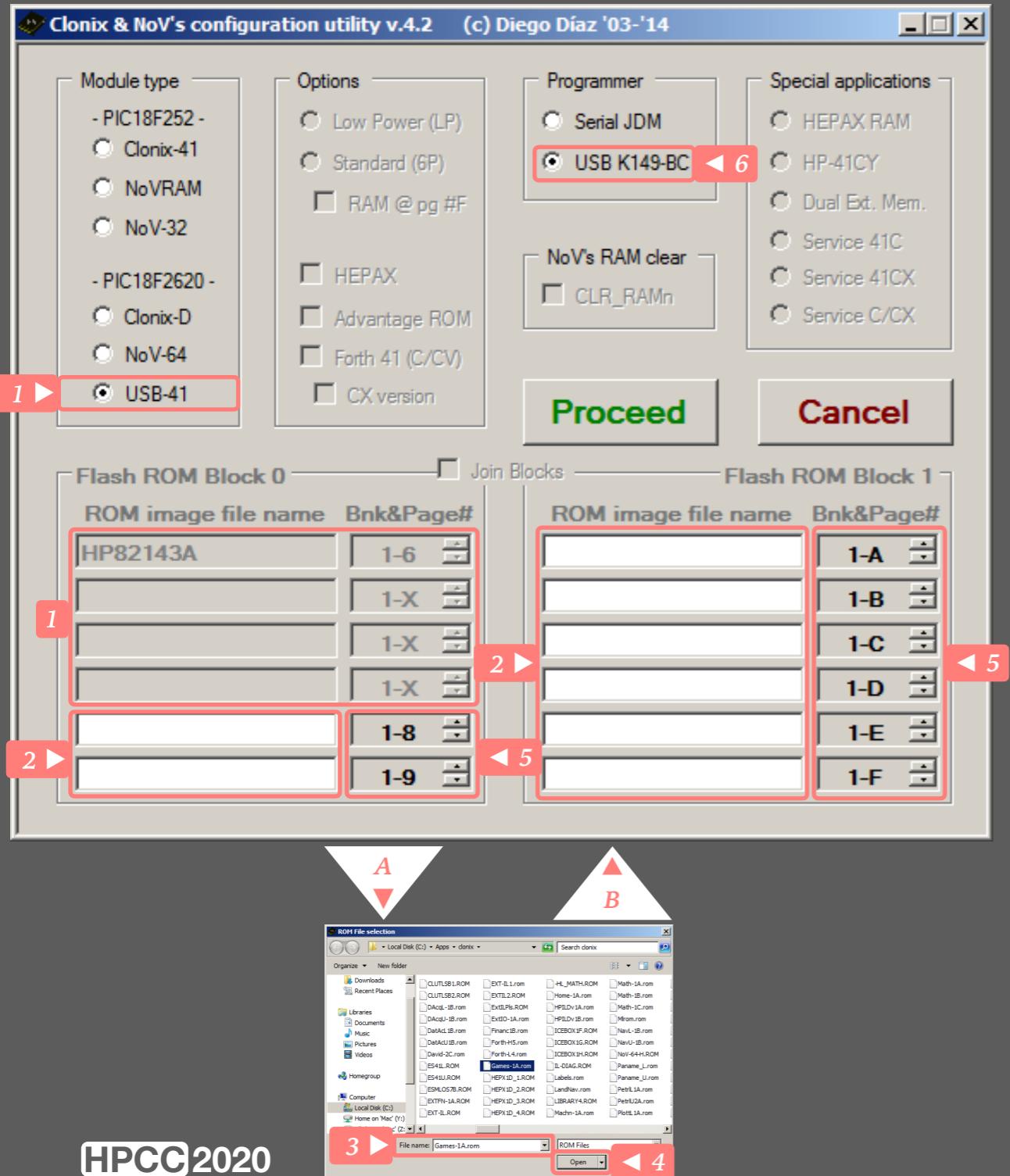
- [HP-82143A](#)



HP-82143A

.....

Goal: loading a modified 82143A printer ROM and optionally other ROMs into the module.



1. Select **USB-41** option.

Load printer ROM image at page #6.

For each ROM file you want to map:

2. Click in one the **ROM image file name** white space to show file selection dialog.
3. Select ROM file name.
4. Click on **Open** button.
5. Select the **Bank [1..4]** & **Page [#4..#F]** you want to map you ROM image to.

Go to Programming section.

PROGRAMMING

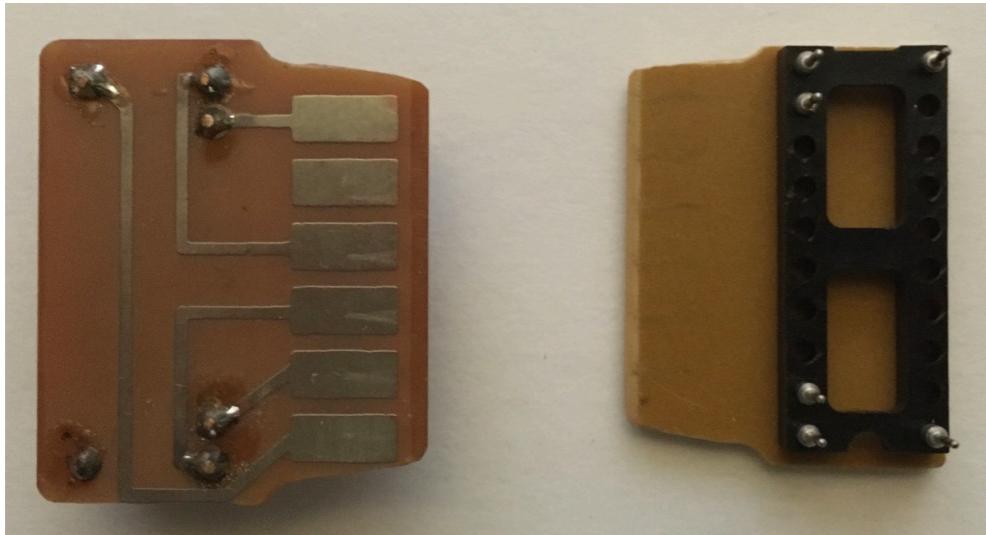
.....

Table of Content

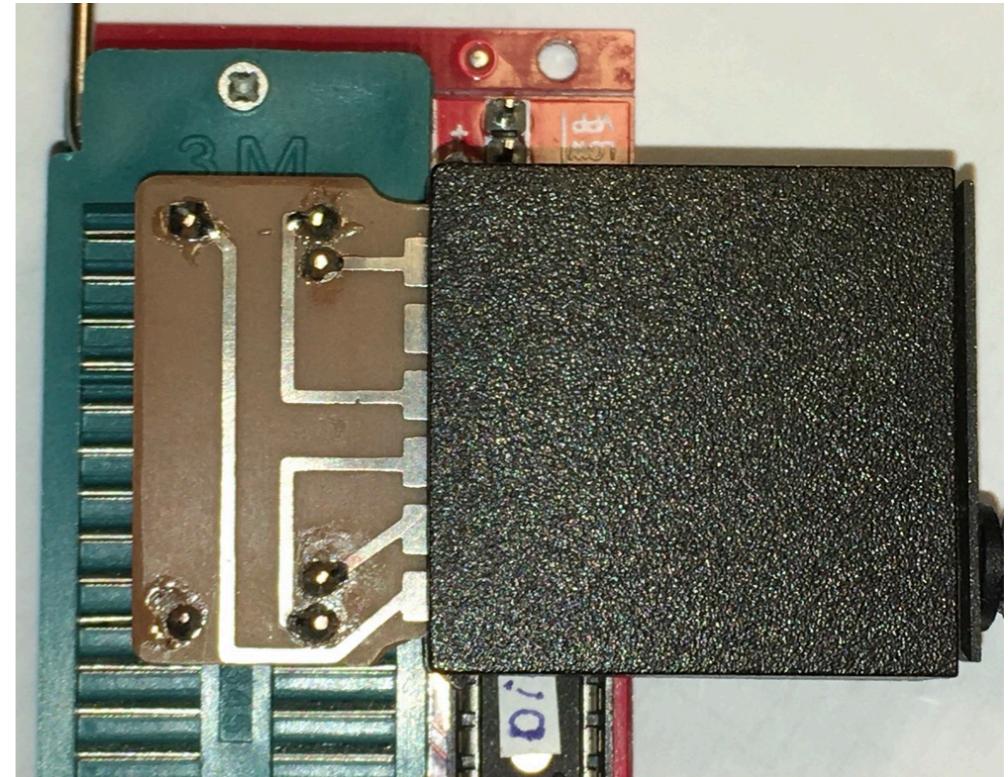
- Hardware 1
- Hardware 2
- Hardware 3
- Software
- PIC Asm & Intel Hex Files
- File Generation Failed
- File Generation Successful
- Programmer Not Found
- File Upload Failed 1
- File Upload Failed 2
- File Upload Successful

Hardware 1 (Before Sept. 2020)

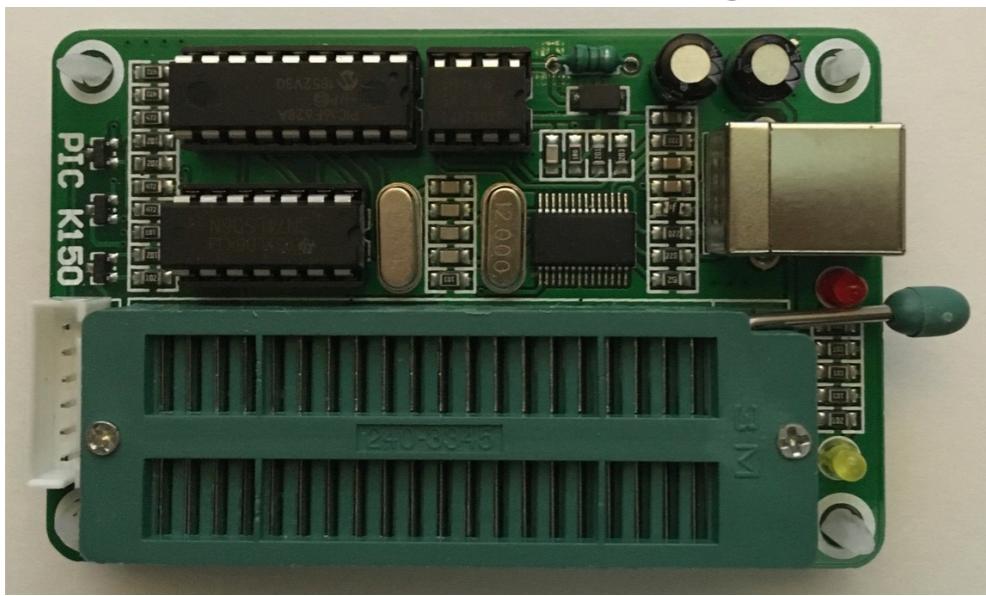
- Module Adapter



- Prog.+Adapter+Module



- K150 USB PIC Programmer



- K149 USB PIC Programmer

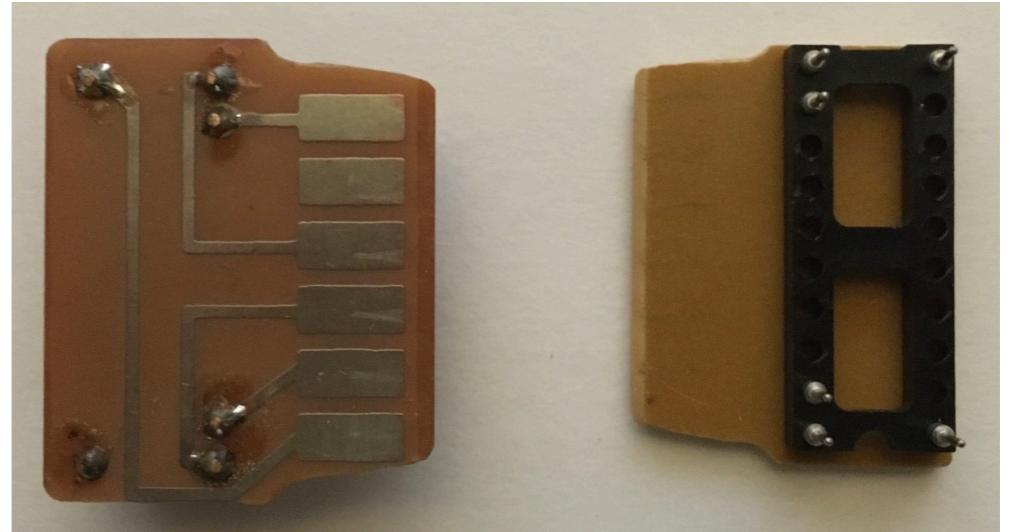


Hardware 2 (Before Sept. 2020)

- Modules & PIC Programmer Adapter by Diego Díaz
- Clonix-D cost is 100 € 
- USB-41 cost is 110 € 
- NoV-64d cost is 140 € 
- Adapter cost is 10 € 
- K150 PIC Programmer + Adapter cost is 30 € 
- Ordering information at www.clonix41.org

Minimum requirement:

K150 USB PIC programmer must have the 18A protocol firmware installed.



Hardware 3 (After Sept. 2020)

- Modules & PIC Programmer Adapter by Diego Díaz

- Clonix-D cost is 100 €



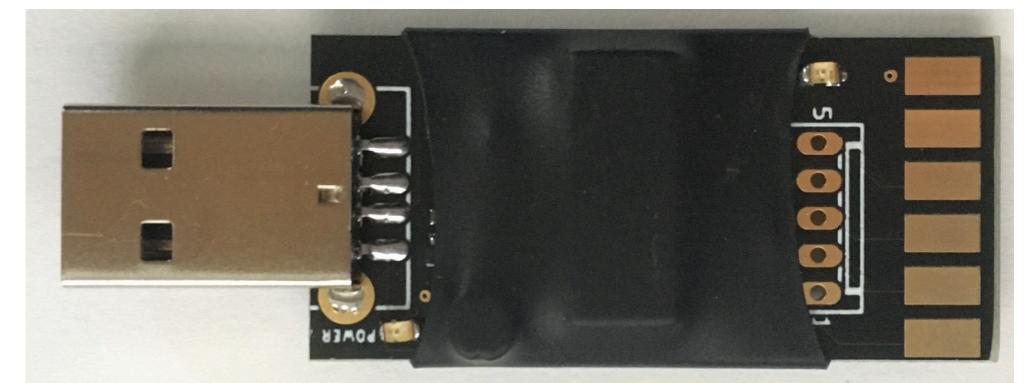
- USB-41 cost is 110 €



- NoV-64d cost is 140 €



- PICKit 2 cost is 40 €



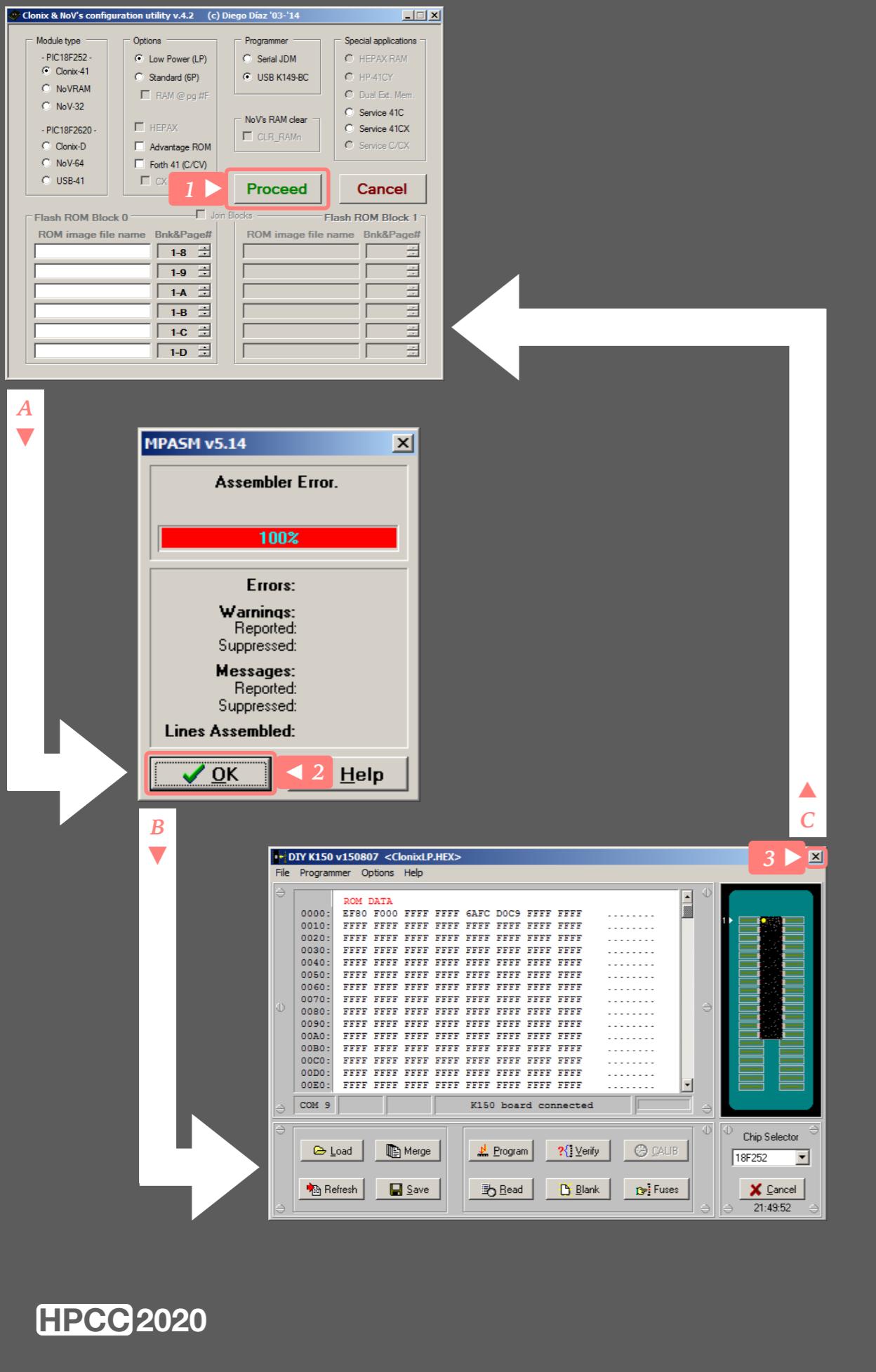
- Ordering information at www.clonix41.org

Software

- Clonix & NoV Configuration Utility v4.2
 - Windows utility written by Diego Díaz to configure Clonix and NoV modules.
 - Read Installation section for software download and installation procedures.
- MPASM (included in Clonix_CD_090315.zip)
 - Microchip Assembler for PIC microcontrollers
 - MPASM User's Guide with MPLINK and MPLIB
ww1.microchip.com/downloads/en/devicedoc/33014g.pdf
 - MPLAB development system
www.microchip.com/mplab/embedded-software-center
- PIC Programming Software (included in Clonix_CD_090315.zip)
 - K150 PIC Programmer Manual
www.sigmaelectronics.net/manuals/K150.pdf
 - Micropro / MicroBurn DIY Software
www.ozitronics.com/download/DIYpack25EP2.zip

PIC Assembly & Intel Hex Files (.asm & .hex)

| Filename | Module(s) | Description |
|----------|----------------------|--|
| CLONIXLP | Clonix 41 | Clonix 41 Low Power |
| CLONIX6P | Clonix 41 | Clonix 41 Standard (6 pages) |
| CLONIX6R | Clonix 41 | Clonix 41 Standard (6 pages) with 512 words RAM |
| CLONIX-D | Clonix-D | Clonix-D |
| CLONIX-P | USB-41 | HP-82143A Printer Simulation |
| CLONIXU4 | USB-41 | Page Transfer Utility : ROMCOPY piggyback firmware |
| NOVRAM-H | NOVRAM | HEPAX Emulation - Version 1D |
| NOV-32-H | NOV-32 | HEPAX Emulation - Version 1D |
| NOV-64-H | NOV-64(d) | HEPAX Emulation - Version 1D |
| NOV-64H4 | NOV-64(d) | HEPAX Emulation - Version 4H (need Library4) |
| CLR_RAM1 | NOVRAM | HEPAX RAM Clear |
| CLR_RAM2 | NOV-32 | HEPAX RAM Clear |
| CLR_RAM4 | NOV-64(d) | HEPAX RAM Clear |
| HPX25RAM | NOVRAM & NOV-32 | HEPAX RAM Emulation |
| HPX26RAM | NOV-64(d) | HEPAX RAM Emulation |
| NOV-64CY | NOV-64(d) | W&W HP-41CY & RAMBOX II Emulation |
| NOV-64XM | NOV-64(d) & Clonix-D | Dual X-Memory Emulation |
| SERVC_C | All except USB-41 | Service 41C |
| SERVC_CX | All except USB-41 | Service 41CX |
| SRVCC_CX | Clonix-D | Service C/CX (even/odd port select service module) |



File Generation Failed

In Clonix & NoV Configuration Utility ...

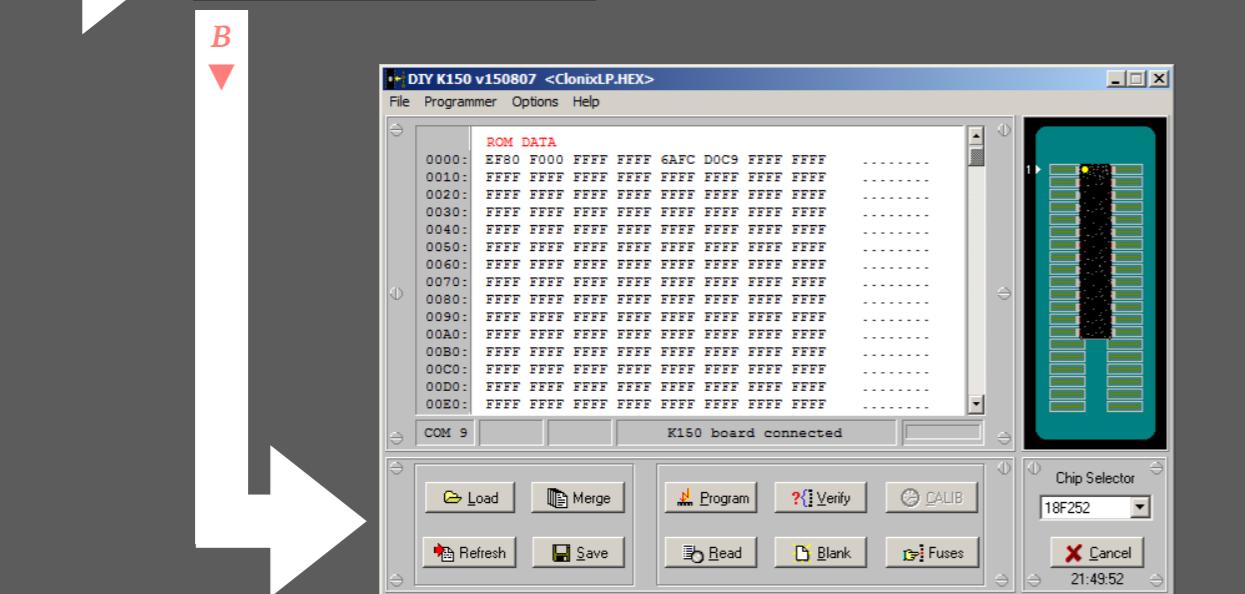
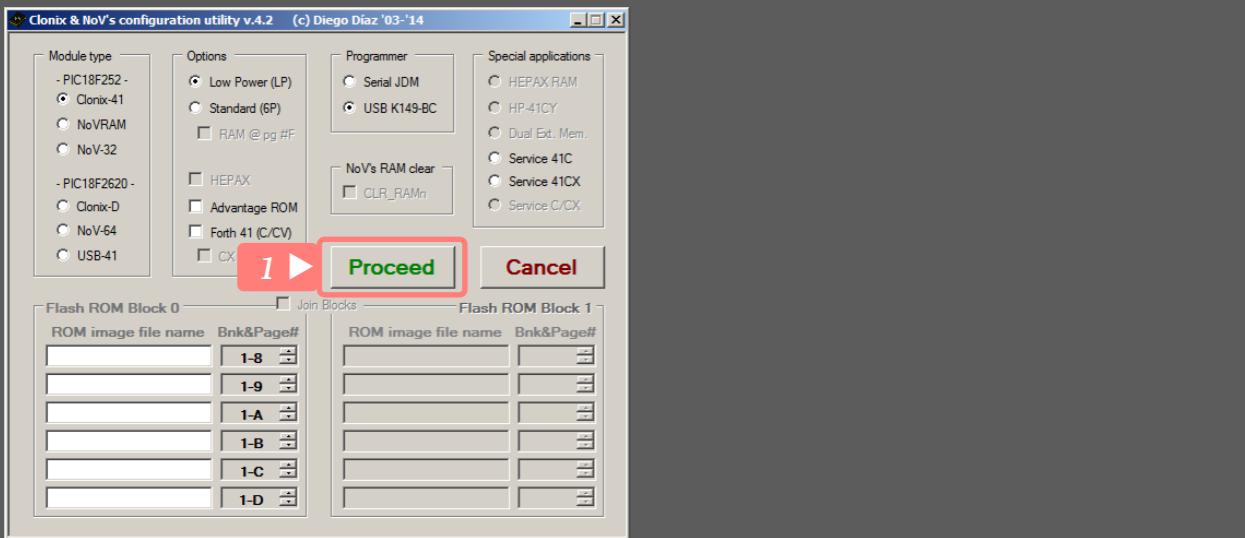
1. Press Proceed to start MPASM.

Assembler was unable to compile the source file, progress bar is red, hex file generation was unsuccessful.

2. Press OK to start MicroBurn (DIY K150) application.

MPASM failed its file generation, we cannot continue further.

3. Close MicroBurn (DIY K150) to go back to Clonix & NoV Configuration Utility application and validate your configuration.



File Generation Successful

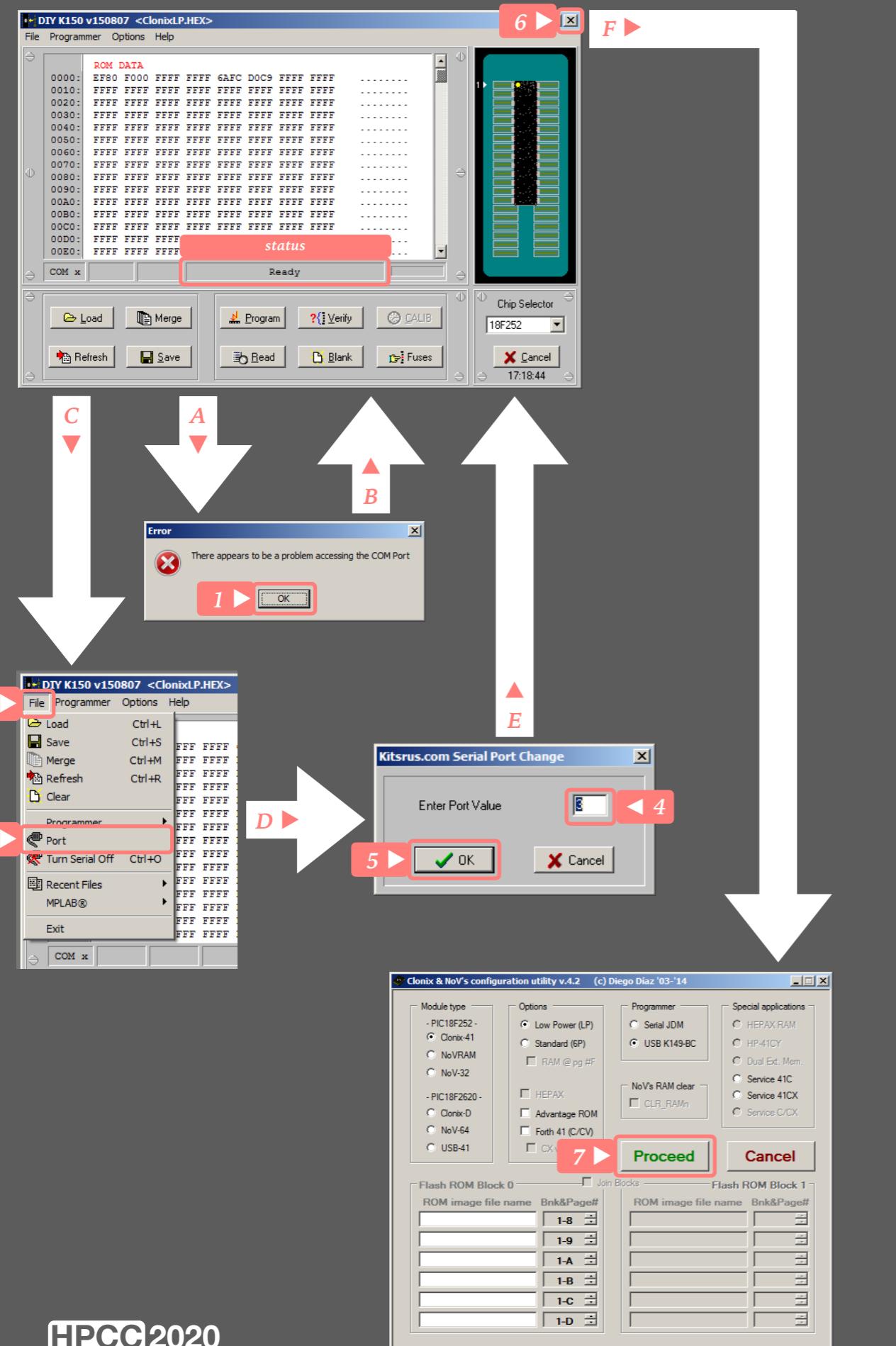
.....

In Clonix & NoV Configuration Utility ...

1. Press Proceed to start MPASM.

Assembler was able to compile the source file, progress bar is green, hex file generation was successful.

2. Press OK to start MicroBurn (DIY K150) application.



Programmer Not Found

In MicroBurn (DIY K150) ...

Upon application start, a communication error dialog box is telling us that the application is not able to communicate with the PIC programmer.

1. Press **OK** to acknowledge the error.

Verify that your RS-232 or USB PIC programmer is correctly connected. Verify that serial communication port number is valid.

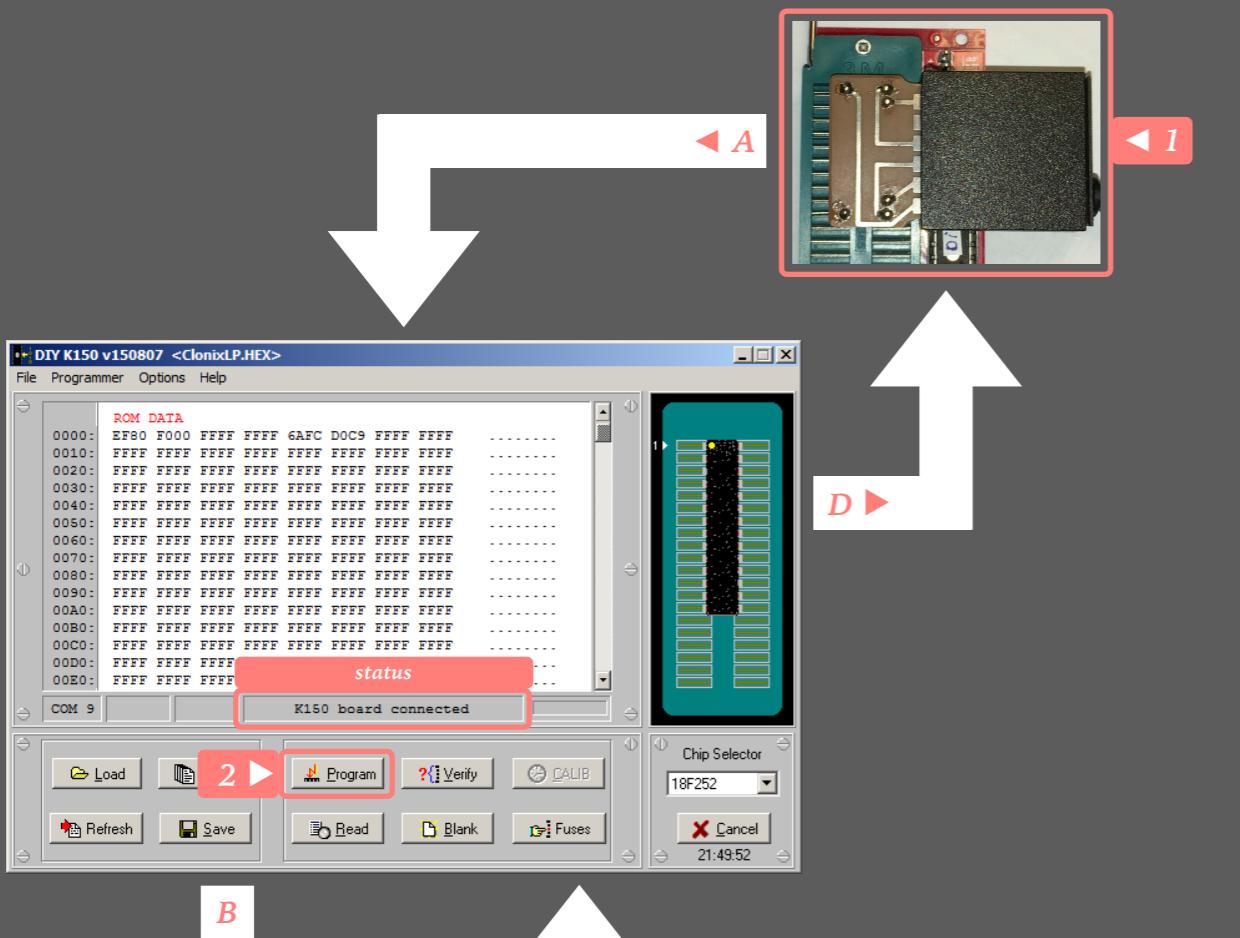
2. Select menu **File**.
3. Select sub-menu **Port**.

*In the **Serial Port Change** dialog box:*

4. Enter PIC Programmer COM Port
5. Press **OK** to accept and close the dialog box.

If connection is established you should see K150 board connected displayed in the status field.

6. Close **MicroBurn (DIY K150)** to go back to **Clonix & NoV Configuration Utility**
7. Press **Proceed** again.



File Upload Failed 1

.....

1. Verify that your module is correctly inserted in the adapter.

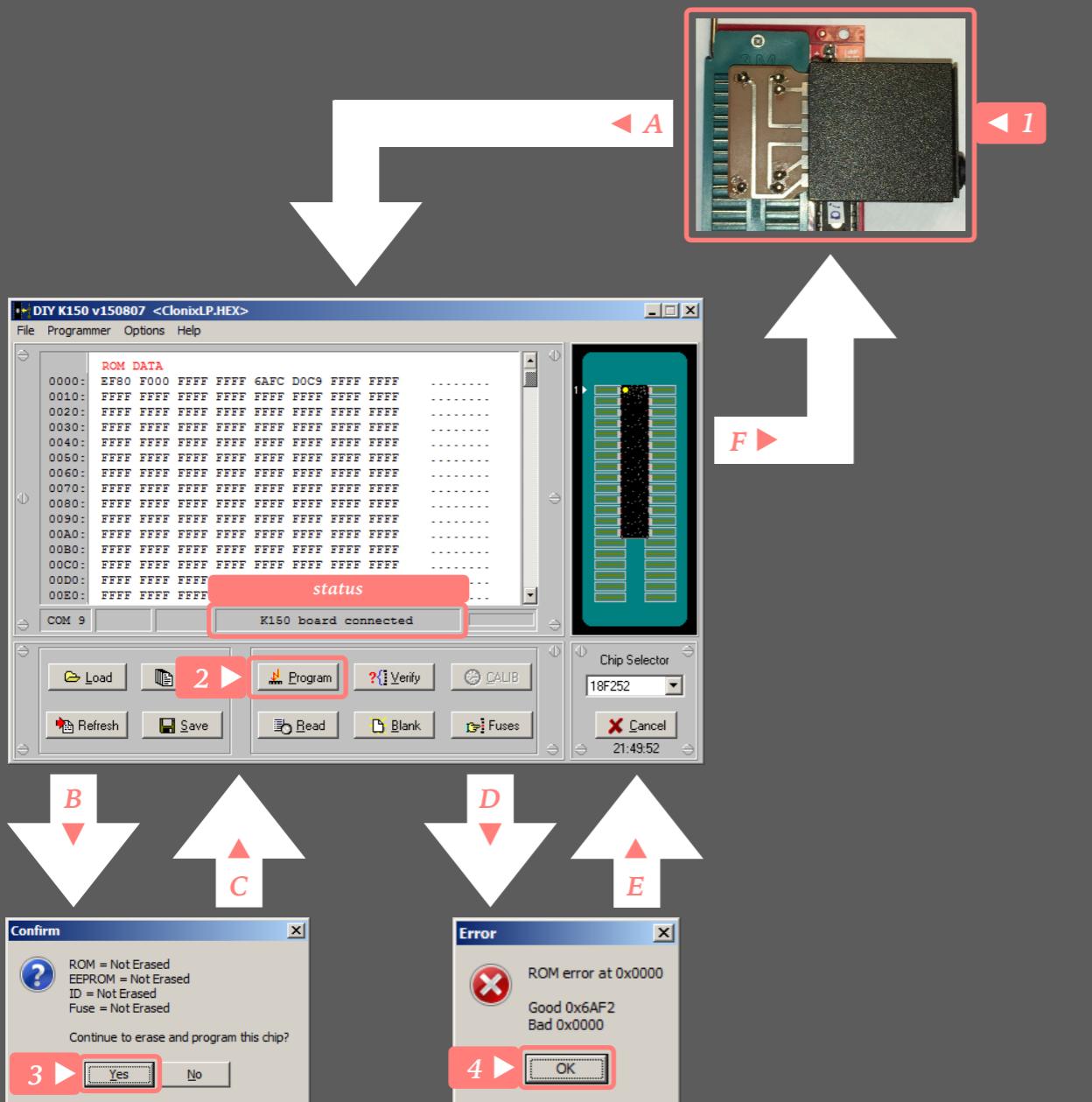
In MicroBurn (DIY K150) ...

2. Press Program to start programming.

A confirm dialog box is telling us that the module is not recognized.

3. Press No in the Confirm dialog box.

Go back to step 1 until successful.



File Upload Failed 2

.....

1. Verify that your module is correctly inserted in the adapter.

In MicroBurn (DIY K150) ...

2. Press Program to start programming.

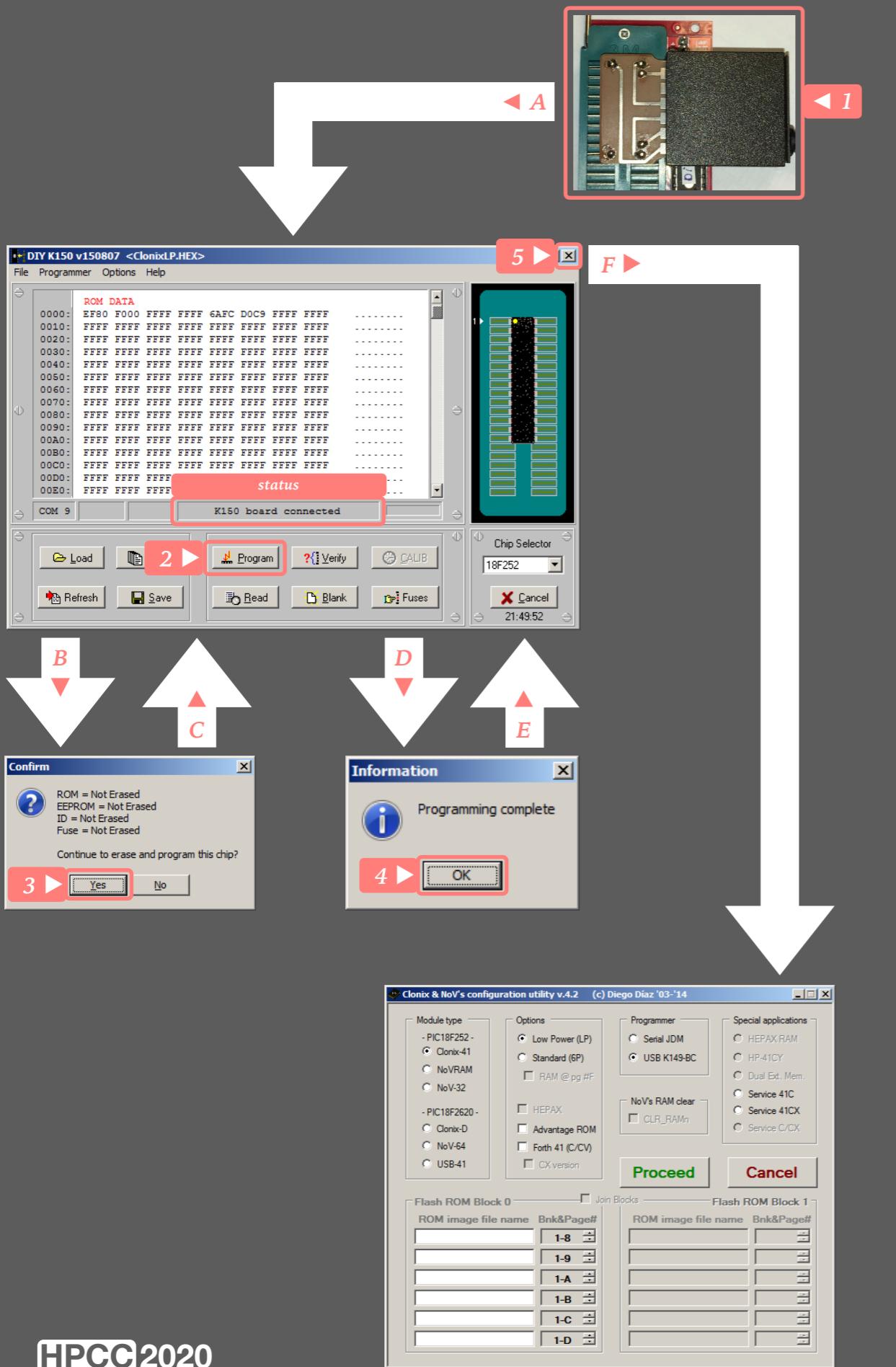
A confirm dialog box is telling us that the module is recognized.

3. Press Yes in the Confirm dialog box.

Several status are displayed in the status field during hex file uploading. Here, module programming failed.

4. Press OK in the Error dialog box.

Go back to step 1 until successful.



File Upload Successful

1. Verify that your module is correctly inserted in the adapter.

In MicroBurn (DIY K150) ...

2. Press Program to start programming.

A *confirm dialog box* is telling us that the module is recognized.

3. Press Yes in the Confirm dialog box.

Several status are displayed in the status field during hex file uploading. Module programming worked.

4. Press OK in the Information dialog box.

Remove your module from the adapter.

5. Close MicroBurn (DIY K150) application to go back to Clonix & NoV Configuration Utility application.

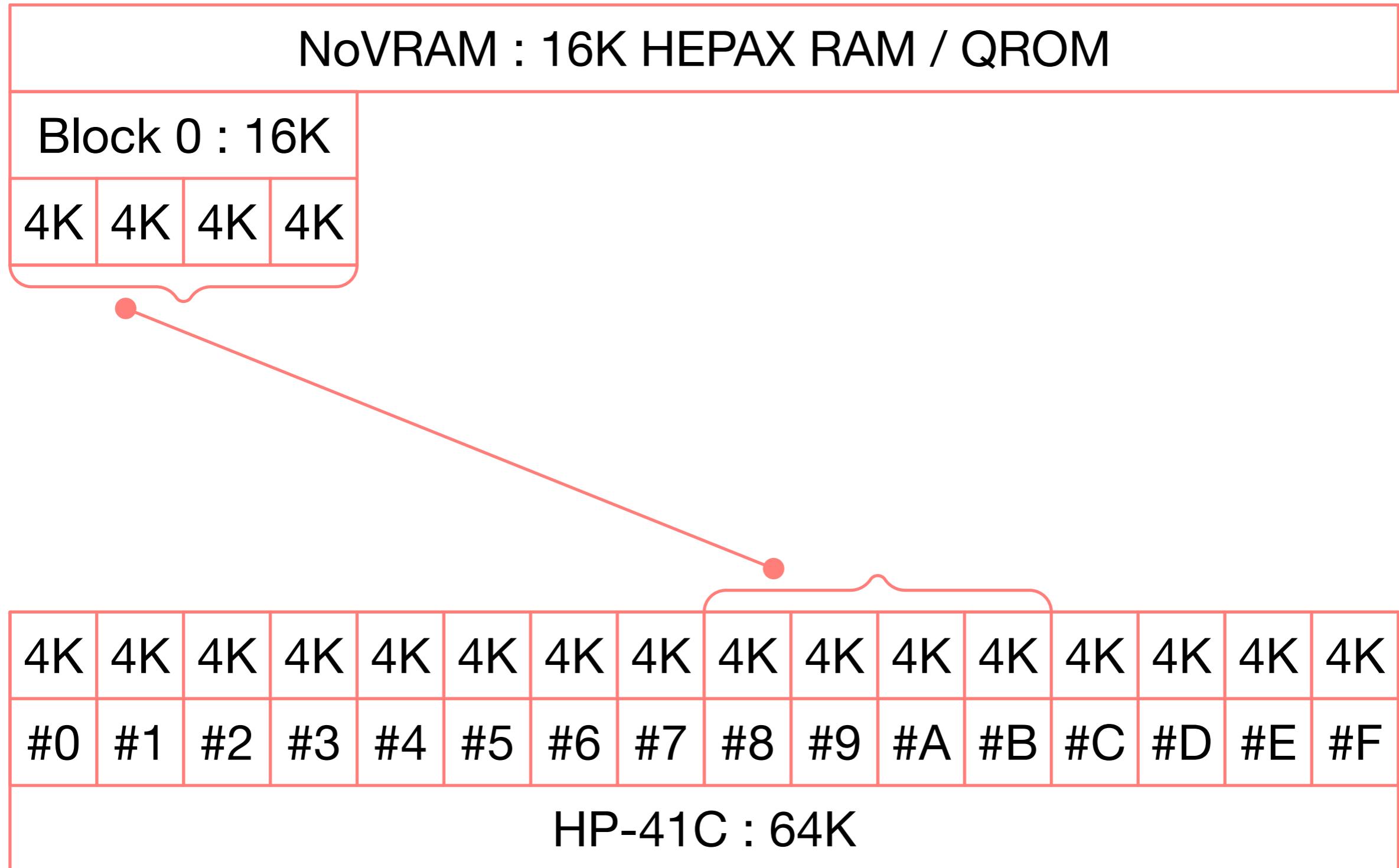
NOV RUNTIME CONFIGURATION

.....

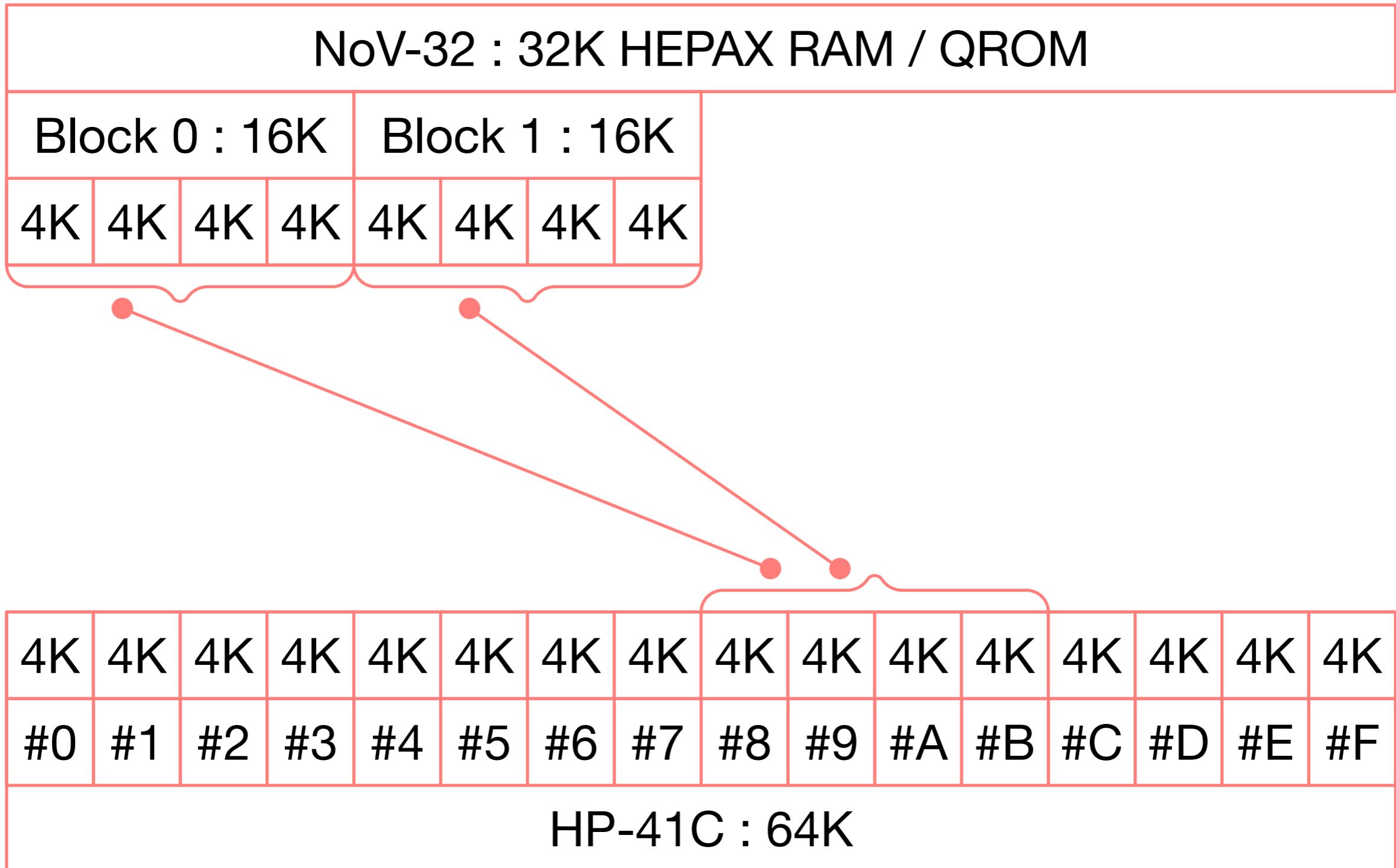
Table of Content

- ▶ 16K RAM Mapping
- ▶ 32K RAM Mapping
- ▶ 64K RAM Mapping
- ▶ 24K Flash Mapping
- ▶ 48K Flash Mapping
- ▶ Control Word
- ▶ Control Word : NoV-32
- ▶ Control Word : NoV-64(d)
- ▶ Crash Recovery Function
- ▶ ROM Shadowing : NoV-64(d)
- ▶ QROM Protection : NoV-64(d)
- ▶ NoV Modes

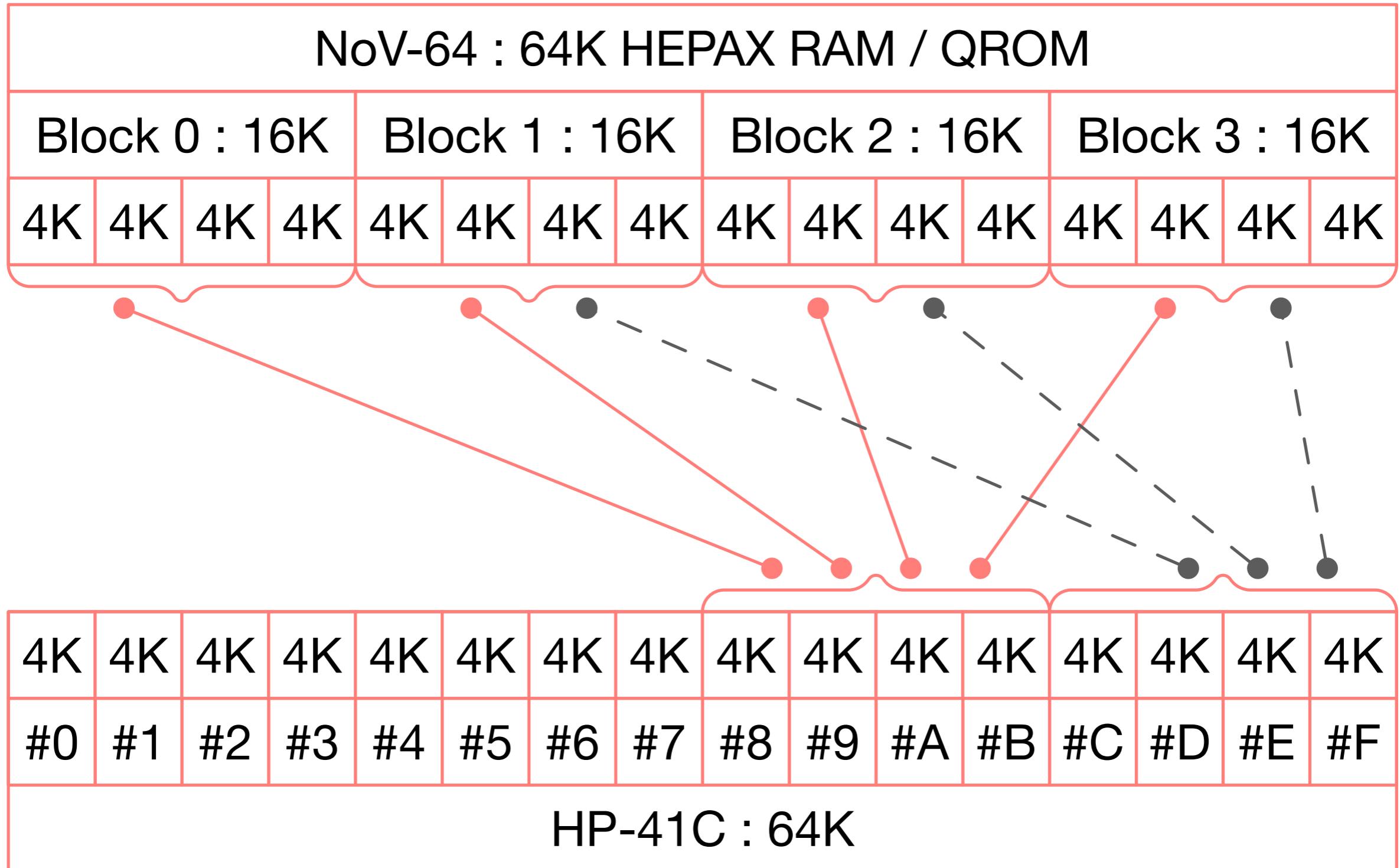
16K RAM Mapping



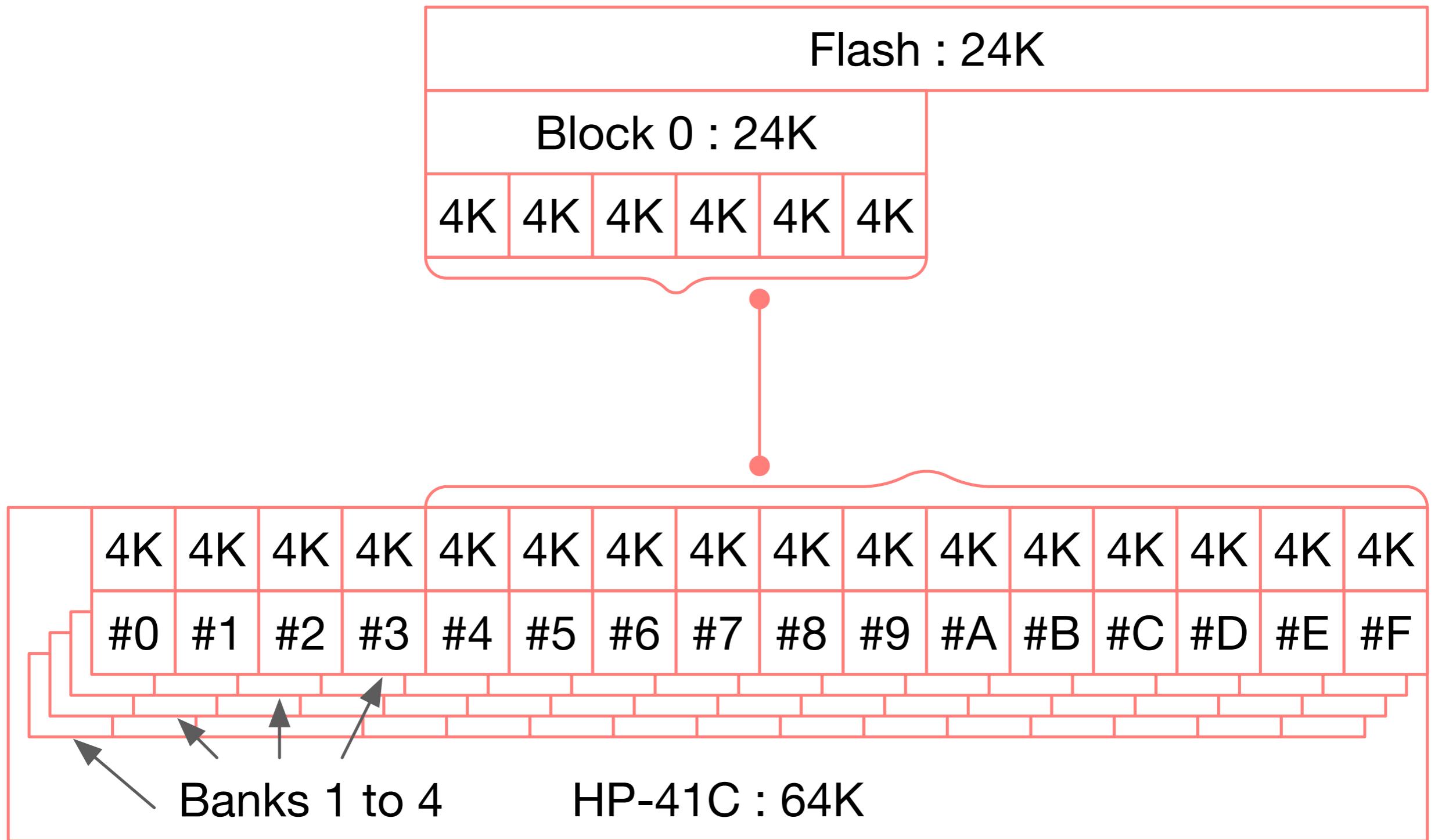
32K RAM Mapping



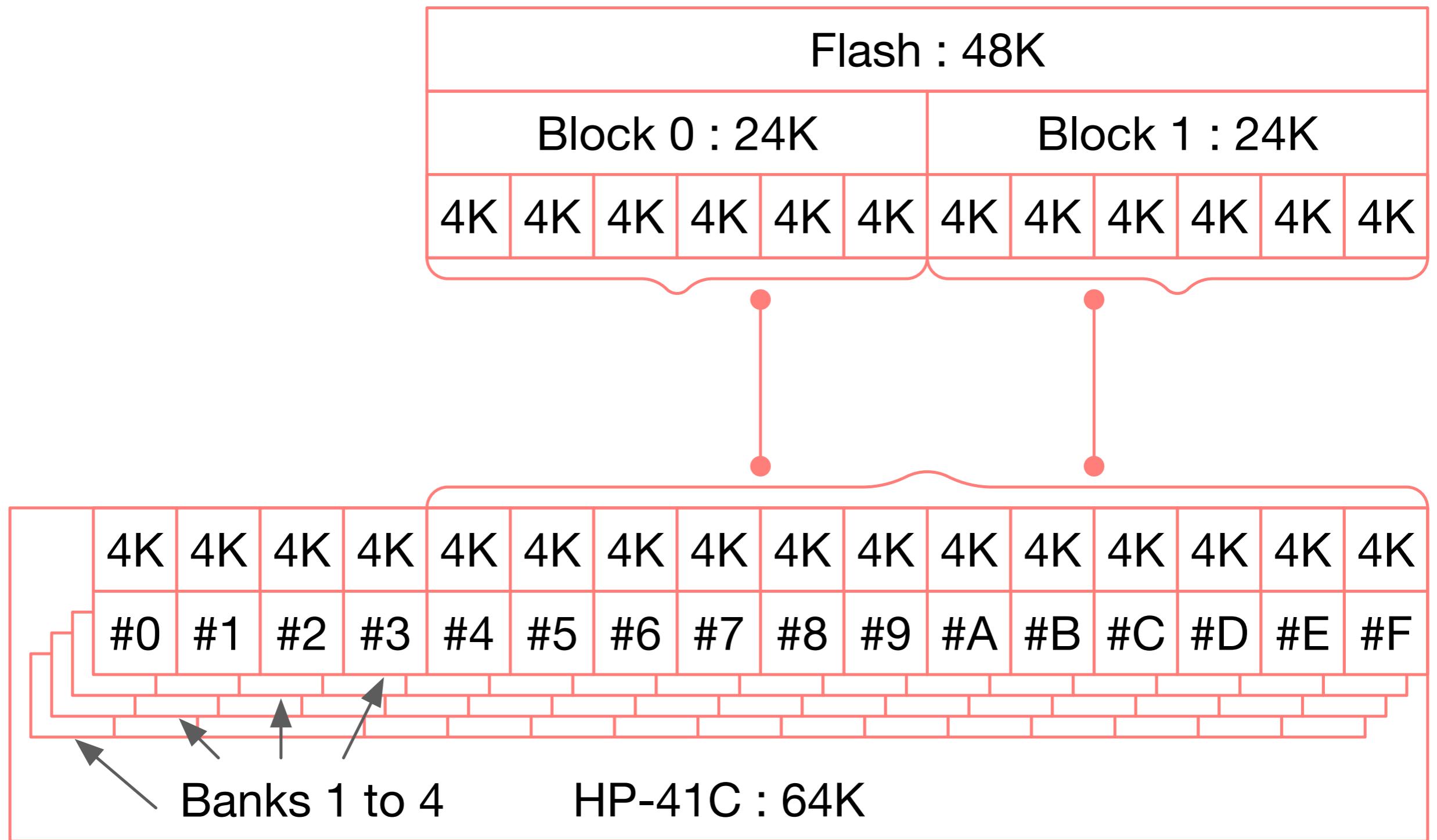
64K RAM Mapping



24K FFlash Mapping



48K FFlash Mapping



Control Word

- NoV-32 & NoV-64(d) modules has the ability to change their configuration at runtime.
- The configuration space (aka Control Word) is located at address 4100.
- Next slides provide the details of what the configuration value means.
- Assuming the module is configured in HEPAX mode, the procedure to change the configuration is ...
 - [XEQ] [ALPHA] HEXEDIT [ALPHA]

You should see: ADR: _____

- Enter 4100

You should see: ADR: 4100 then 4100 CCC _____ (CCC is the current configuration value).

- Enter the new configuration value: NNN

You should see: 4100 CCC NNN then next address location 4101 ??? _____

- [←]

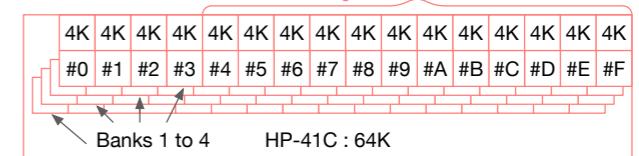
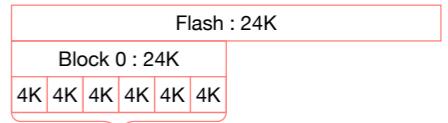
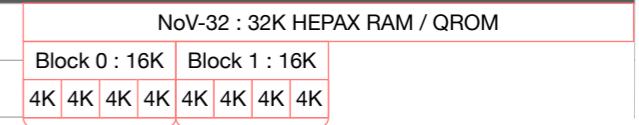
You should see: ADR: _____

- [←] to go back to normal mode.
- The new configuration is now active for the RAM part, but a power cycle is needed for the Flash configuration to become active (if modified).

Control Word : NoV-32

9 8 7 6 5 4 3 2 1 0 Hex Description

| | |
|--------------------------|---|
| 0 0 0 0 0 0 0 0 0 0 #000 | RAM Block 0 mapped to pages #8 to #B in bank 1, 16K HEPAX RAM |
| 0 0 0 0 0 0 0 0 0 1 #001 | RAM Block 1 mapped to pages #8 to #B in bank 1, 16K HEPAX RAM |



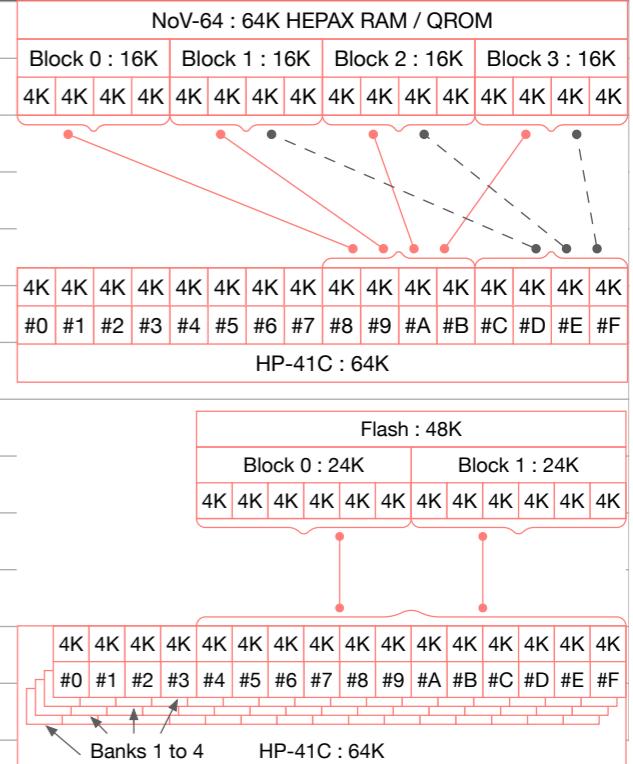
0 0 0 0 0 0 0 0 0 #000 Value at module insertion

9 8 7 6 5 4 3 2 1 0 Examples

| | |
|--------------------------|---|
| 1 1 1 1 1 1 1 1 1 0 #000 | 16K HEPAX RAM [block 0, pages #8..#B, bank 1] |
| 1 0 1 0 1 0 1 0 0 1 #001 | 16K HEPAX RAM [block 1, pages #8..#B, bank 1] |

Control Word : NoV-64(d)

| 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Hex | Description |
|---|---|---|---|---|---|---|---|---|------|--|---|
| | | | | | 0 | 0 | 0 | 0 | #xx0 | 1 st 16K HEPAX RAM mapped [block 0, pages #8..#B, bank 1] | |
| | | | | | 0 | 0 | 0 | 1 | #xx1 | 1 st 16K HEPAX RAM mapped [block 1, pages #8..#B, bank 1] | |
| | | | | | 0 | 0 | 1 | 0 | #xx2 | 1 st 16K HEPAX RAM mapped [block 2, pages #8..#B, bank 1] | |
| | | | | | 0 | 0 | 1 | 1 | #xx3 | 1 st 16K HEPAX RAM mapped [block 3, pages #8..#B, bank 1] | |
| | | | | 0 | 0 | 0 | 0 | 0 | #x0x | 2 nd 16K HEPAX RAM unmapped | |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | | | #01x | 2 nd 16K HEPAX RAM mapped [block 1, pages #C..#F, bank 1] (Flash must be unmapped) | |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | | | #02x | 2 nd 16K HEPAX RAM mapped [block 2, pages #C..#F, bank 1] (Flash must be unmapped) | |
| 0 | 0 | 0 | 0 | 1 | 1 | | | | #03x | 2 nd 16K HEPAX RAM mapped [block 3, pages #C..#F, bank 1] (Flash must be unmapped) | |
| 0 | 0 | | | | | | | | #0xx | 16K Flash unmapped | |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | | | #10x | 16K Flash mapped [block 0, any unused pages/bank] (2 nd 16K HEPAX RAM must be unmapped) | |
| 1 | 0 | 0 | 0 | 0 | 0 | | | | #20x | 16K Flash mapped [block 1, any unused pages/bank] (2 nd 16K HEPAX RAM must be unmapped) | |
| 1 | 1 | 0 | 0 | P | P | 0 | 0 | B | B | #3PB | Copy RAM page PP (0..3) in block BB (0..3) to Flash page 5 in block 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | #3FF | Erase Flash page 5 in block 1 | |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | #100 | Value at module insertion | |



| 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Examples |
|---|---|---|---|---|---|---|---|---|------|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | #000 | No Flash, 16K HEPAX RAM [block 0, pages #8..#B, bank 1] |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | | #001 | No Flash, 16K HEPAX RAM [block 1, pages #8..#B, bank 1] |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | #002 | No Flash, 16K HEPAX RAM [block 2, pages #8..#B, bank 1] |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | #003 | No Flash, 16K HEPAX RAM [block 3, pages #8..#B, bank 1] |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | #020 | No Flash, 32K HEPAX RAM [block 2, pages #C..#F, bank 1] & [block 0, pages #8..#B, bank 1] |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | #031 | No Flash, 32K HEPAX RAM [block 3, pages #C..#F, bank 1] & [block 1, pages #8..#B, bank 1] |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | #102 | 16K Flash [block 0], 16K HEPAX RAM [block 2, pages #8..#B, bank 1] |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | #203 | 16K Flash [block 1], 16K HEPAX RAM [block 3, pages #8..#B, bank 1] |

Crash Recovery Function

- The goal of this feature is to put the module in safe/recovery mode.
 - Great to get out of a dead lock situation when you have a polling point bug in your mcode.
 - When a corrupted 4K RAM/QRAM page is creating a calculator lockup.
 - Works on all NoV modules. (NoVRAM, NoV-32, NoV-64 & NoV-64d)
- When this mode is activated:
 - HEPAX RAM/QROM read is disabled.
Port catalog [HEPAX 002] no longer shows these pages.
 - HEPAX RAM/QROM write is enabled.
Allowing you to clear or to overwrite the content of pages #8 to #B.
 - HEPAX ROM is mapped to page #C.
Temporary overwriting Flash mapping for that page.
 - Control Word 4100 configuration is unchanged
- Manual Activation:
 - In OFF mode, hold [ENTER] key down and press [ON] key twice in quick succession.
 - Not working when inside a 41CL.
- Automatic Activation:
 - CRF is automatically enabled after a memory lost. (Including the 41CL)
- Validation:
 - #1: XEQ "HEPAX" then 002 to execute a port catalog.
You should see that ports #8 to #B are empty and that HEPAX ROM is mapped to port #C.
 - #2: XEQ "HEPDIR" to list HEPAX RAM content.
If successful, "H:NO FILESYS" should be displayed on the screen.
- Deactivation:
 - Do a power cycle: [ON][ON]

ROM Shadowing & QROM Protection : NoV-64(d)

- RAM & ROM Shadowing
 - When a physical module is inserted and a page conflict arise with the NoV configuration, the firmware give precedence to the physical module and temporary unmapped the page from the NoV module.
 - *Warning: if the physical module page address is in conflict with a HEPAX RAM page, you should manage the issue otherwise you may lose some files or the entire HEPAX filesystem.*
- QROM Protection
 - The NoV module fully support HEPAX RAM write protection.
 - Usage: X must contain the page number to be protected (8..15) then you execute RAMTOG to activate or deactivate write protection.
 - *Warning: never activate write protection on a HEPAX RAM filesystem page.*

NoV Notes

- NoV configured as Clonix:
 - Control word not available.
- NoV configured as NoV:
 - Control word value manage module memory mapping & behavior.
 - Control word value is lost when module is unplugged.
- NoV-64 configured as Clonix-D:
 - Odd/even port sensing not working.
 - Blocks unmerged: 24K usable. (*Flash ROM Block 0 must match Flash ROM Block 1.*)
 - Blocks merged: 48K usable.
- NoV-64d configured as Clonix-D:
 - Odd/even port sensing working.
 - Blocks unmerged: 2 x 24K usable.
 - Blocks merged: 48K usable.

HEPAX RAM CLEARING

Table of Content

- HEPAX vs NoV Modules
- Clearing with Config. Utility
- Manual Clearing

HEPAX vs NoV Modules

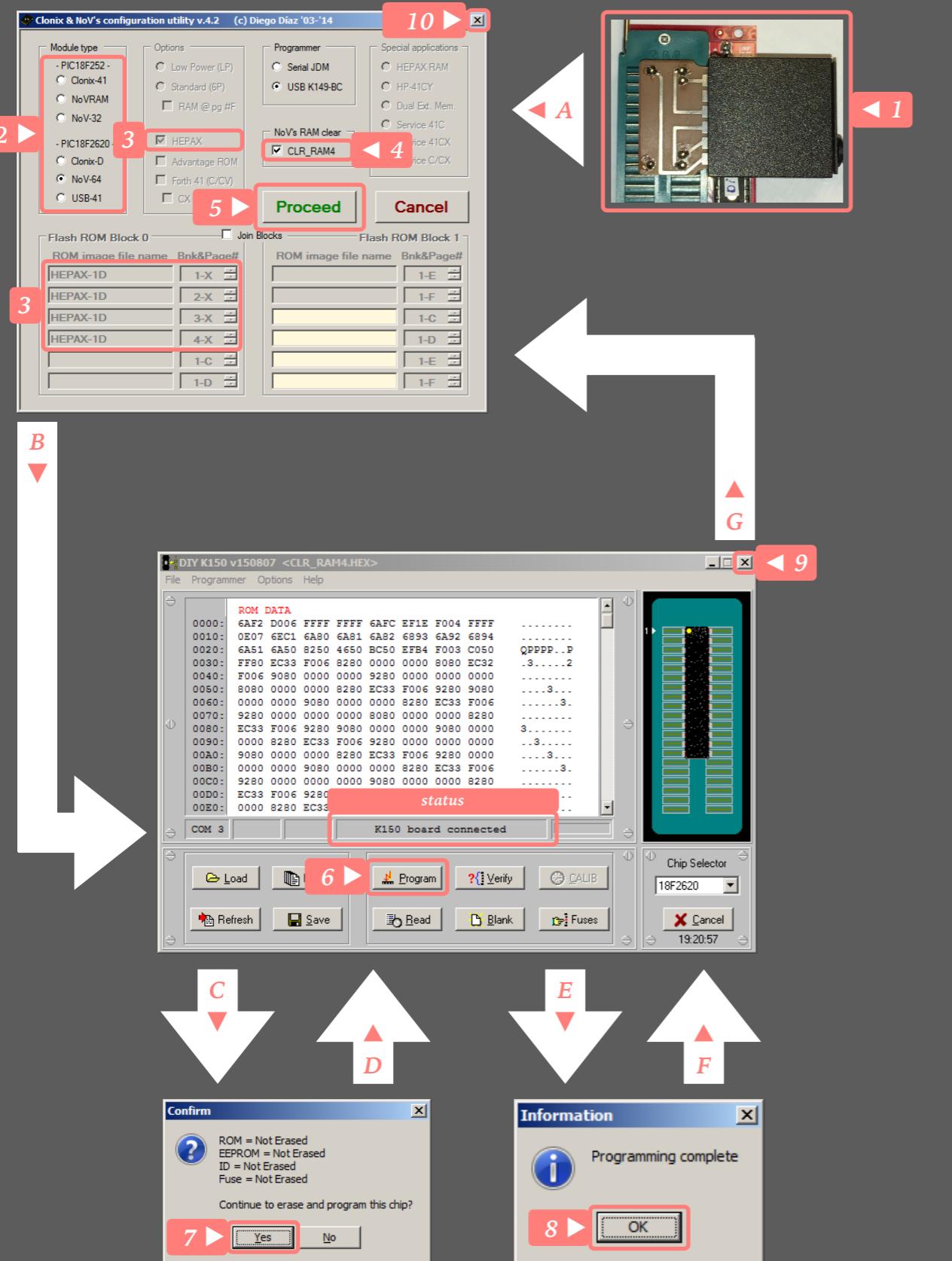
- The RAM type used in the original HEPAX module was SRAM, so to clear its content, you simply had to remove the module from the calculator, wait a bit, reinsert it back and it was cleared.
- The RAM type used in the NoV modules is FRAM, the benefit of this technology is that it keep its content even when unplugged. The downside of it, is that the above procedure no longer works.
- The next slides shows how to clear NoV HEPAX RAM, they assume that your module is configured as HEPAX.

CLEARING WITH CONFIGURATION UTILITY

*This method is here for completeness sake,
it has been proven to be unreliable, please
use manual clearing for better results.*

Table of Content

- Configuration
- Clearing



Configuration

Goal: loading clear HEPAX RAM firmware into NoV module.

1. Verify that your module is correctly inserted in the adapter.

In Clonix & NoV Configuration Utility:

2. Select the **NoV** module that match you're module.
3. **HEPAX** is automatically selected but unused.
4. Select **CLR_RAMx** to load a specialized firmware that will clear HEPAX RAM.

5. Press **Proceed** to start MPASM.

In MicroBurn (DIY K150):

6. Press **Program** to start programming.

A *confirm dialog box is telling us that the module is recognized.*

7. Press **Yes** in the **Confirm** dialog box.

Several status are displayed in the status field during hex file uploading. A information dialog box is telling us that the module had been successfully programmed.

8. Press **OK** in the **Information** dialog box.

Remove your module from the adapter

9. Close MicroBurn (DIY K150) application to go back to Clonix & NoV Configuration Utility application.

10. Close Clonix & NoV Configuration Utility.

Clearing

- Make sure the calculator is off.
- Insert you're NoV module into any port.
- Do NOT press the [ON] button.
- Wait for about 25 seconds for the erasing procedure to complete.
- **CLR OK** message will be displayed if clearing has been successful.
- **NO CLR** message will be displayed if clearing has failed.

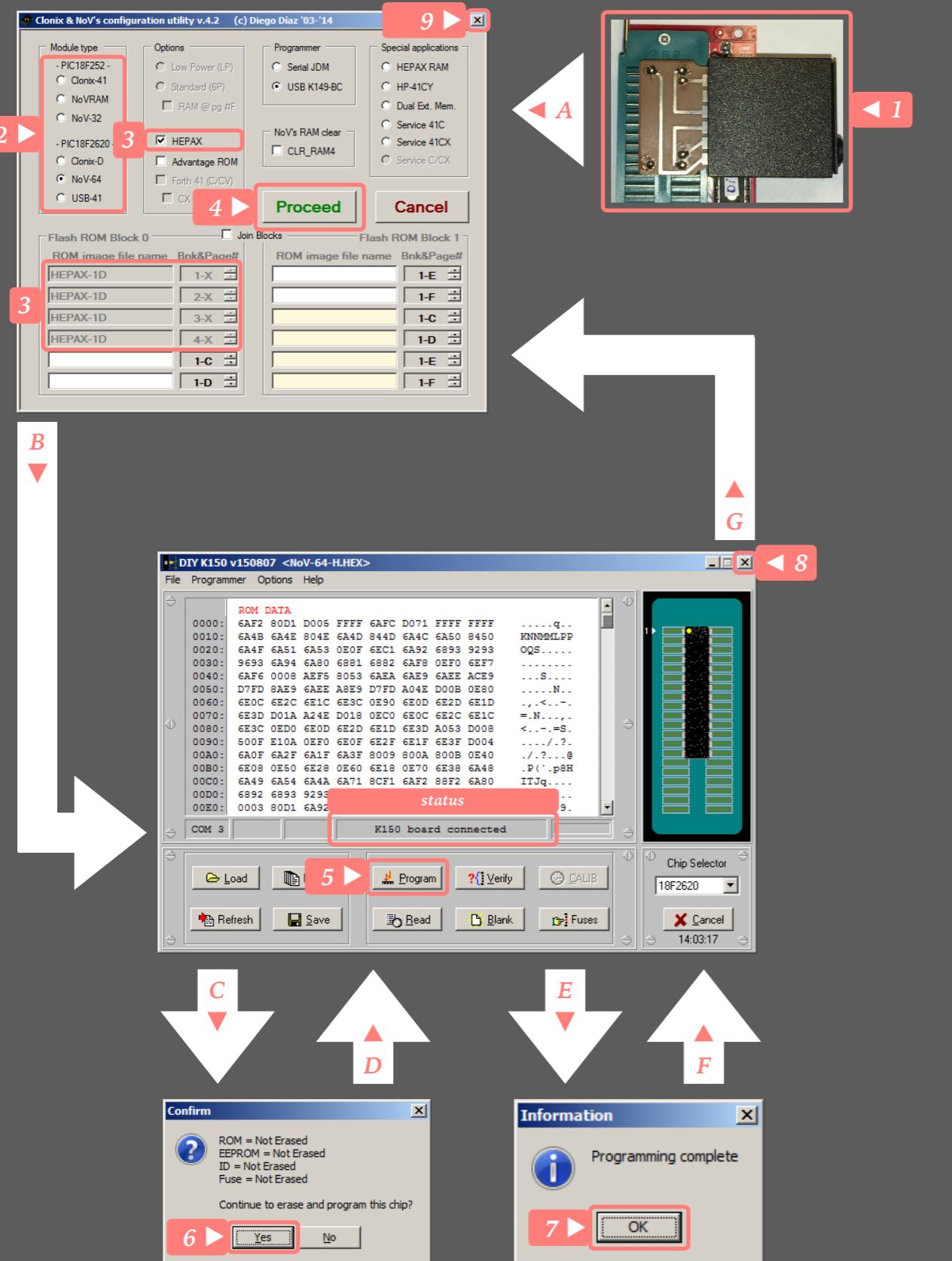
*If you consistently get **NO CLR** message then use one of the other clear methods.*

MANUAL CLEARING

*Works in normal mode
and in crash recovery mode*

Table of Content

- Configuration
- FOCAL Program
- Clearing : NoVRAM & NoV-32
- Clearing : NoV-64(d) [16K]
- Clearing : NoV-64(d) [32K]



Configuration

Goal: loading default HEPAX firmware into NoV module.

1. Verify that your module is correctly inserted in the adapter.

In Clonix & NoV Configuration Utility:

2. Select the **NoV** module that match you're module.
3. **HEPAX** is automatically selected.
4. Press **Proceed** to start MPASM.

In MicroBurn (DIY K150):

5. Press **Program** to start programming.

A confirm dialog box is telling us that the module is recognized.

6. Press **Yes** in the **Confirm** dialog box.

Several status are displayed in the status field during hex file uploading. A information dialog box is telling us that the module had been successfully programmed.

7. Press **OK** in the **Information** dialog box.

Remove your module from the adapter

8. Close **MicroBurn (DIY K150)** application to go back to **Clonix & NoV Configuration Utility** application.
9. Close **Clonix & NoV Configuration Utility**.

FOCAL Program

| Program | Description |
|--------------|---|
| LBL "HCLR16" | HEPAX Clear RAM 16K configuration. |
| 8.011 | Clear page 8 to 11 inclusively. |
| GTO 00 | Goto clear common code. |
| LBL "HCLR32" | HEPAX Clear RAM 32K configuration. |
| 8.015 | Clear page 8 to 15 inclusively. |
| LBL 00 | Clear common code. |
| "OK" | CLRAM confirmation. |
| LBL 01 | Clear loop. |
| VIEW X | Show which page is being cleared. |
| CLRAM | Clear page specified in X. |
| ISG X | Have we finish clearing ? |
| GTO 01 | No, go clear another page. |
| SF 11 | Set autoexec flag. |
| OFF | Rebuild HEPAX pages tags (press ON to complete execution). |
| HEPDIR | Rebuild HEPAX pages links. |
| END | Program end. → X should have 2610 (16K cfg) or 5222 (32K cfg) |

Clearing : NoVRAM & NoV-32

- Clearing a **NoVRAM** module (16K):
 - XEQ "HCLR16"
you should see "H:DIR EMPTY"
and have 2610 free HEPAX reg. in X.
- Clearing a **NoV-32** module (16K):
 - Clearing HEPAX RAM Block 0:
 - If CRF needed, do the CRF procedure below.
 - [XEQ] [ALPHA] HEXEDIT [ALPHA]
 - 4100 then **000** then [\leftarrow] and [\leftarrow]
new configuration is now active.
 - XEQ "HCLR16"
you should see "H:DIR EMPTY"
and have 2610 free HEPAX reg. in X.
- Clearing HEPAX RAM Block 1:
 - If CRF needed, do the CRF procedure below.
 - [XEQ] [ALPHA] HEXEDIT [ALPHA]
 - 4100 then **001** then [\leftarrow] and [\leftarrow]
new configuration is now active.
- XEQ "HCLR16"
you should see "H:DIR EMPTY"
and have 2610 free HEPAX reg. in X.

*CRF Activation: Power off, hold [ENTER] key down and press [ON] key twice in quick succession
CRF Validation: XEQ "HEPDIR" should display "H:NO FILESYS", if not, redo the CRF Activation.*

Clearing : NoV-64(d) [16K HEPAX RAM Mapped]

- Clearing HEPAX RAM Block 0:
 - If CRF needed, do the CRF procedure below.
 - [XEQ] [ALPHA] HEXEDIT [ALPHA]
 - 4100 then 000 then [\leftarrow] and [\leftarrow]
new configuration is now active
 - XEQ "HCLR16"
you should see "H:DIR EMPTY"
and have 2610 free HEPAX reg. in X.
- Clearing HEPAX RAM Block 1:
 - If CRF needed, do the CRF procedure below.
 - [XEQ] [ALPHA] HEXEDIT [ALPHA]
 - 4100 then 001 then [\leftarrow] and [\leftarrow]
new configuration is now active
 - XEQ "HCLR16"
you should see "H:DIR EMPTY"
and have 2610 free HEPAX reg. in X.
- Clearing HEPAX RAM Block 2:
 - If CRF needed, do the CRF procedure below.
 - [XEQ] [ALPHA] HEXEDIT [ALPHA]
 - 4100 then 002 then [\leftarrow] and [\leftarrow]
new configuration is now active
 - XEQ "HCLR16"
you should see "H:DIR EMPTY"
and have 2610 free HEPAX reg. in X.
- Clearing HEPAX RAM Block 3:
 - If CRF needed, do the CRF procedure below.
 - [XEQ] [ALPHA] HEXEDIT [ALPHA]
 - 4100 then 003 then [\leftarrow] and [\leftarrow]
new configuration is now active
 - XEQ "HCLR16"
you should see "H:DIR EMPTY"
and have 2610 free HEPAX reg. in X.

CRF Activation: Power off, hold [ENTER] key down and press [ON] key twice in quick succession
CRF Validation: XEQ "HEPDIR" should display "H:NO FILESYS", if not, redo the CRF Activation.

Clearing : NoV-64(d) [32K HEPAX RAM Mapped]

- There are multiple configuration possible here:
 - Config Pair #X (Double Block 0 & Double Block 1).
 - Config Pair #1 (DB0: **010** & DB1: **032**).
 - Config Pair #2 (DB0: **020** & DB1: **031**).
 - Config Pair #3 (DB0: **030** & DB1: **021**).
- Clearing HEPAX RAM Config Pair #X (Double Block 0):
 - If CRF needed, do the CRF procedure below.
 - [XEQ] [ALPHA] HEXEDIT [ALPHA]
 - **4100** then **010** or **020** or **030** then [] and [] new configuration is now active.
 - XEQ "HCLR32"
you should see "H:DIR EMPTY"
and have 5222 free HEPAX reg. in X.
- Clearing HEPAX RAM Config Pair #X (Double Block 1):
 - If CRF needed, do the CRF procedure below.
 - [XEQ] [ALPHA] HEXEDIT [ALPHA]
 - **4100** then **032** or **031** or **021** then [] and [] new configuration is now active.
 - XEQ "HCLR32"
you should see "H:DIR EMPTY"
and have 5222 free HEPAX reg. in X.

*CRF Activation: Power off, hold [ENTER] key down and press [ON] key twice in quick succession
CRF Validation: XEQ "HEPDIR" should display "H:NO FILESYS", if not, redo the CRF Activation.*

CLOSING TOPICS

Table of Content

- [Review](#)
- [Questions & Answers](#)
- [Index](#)

Review

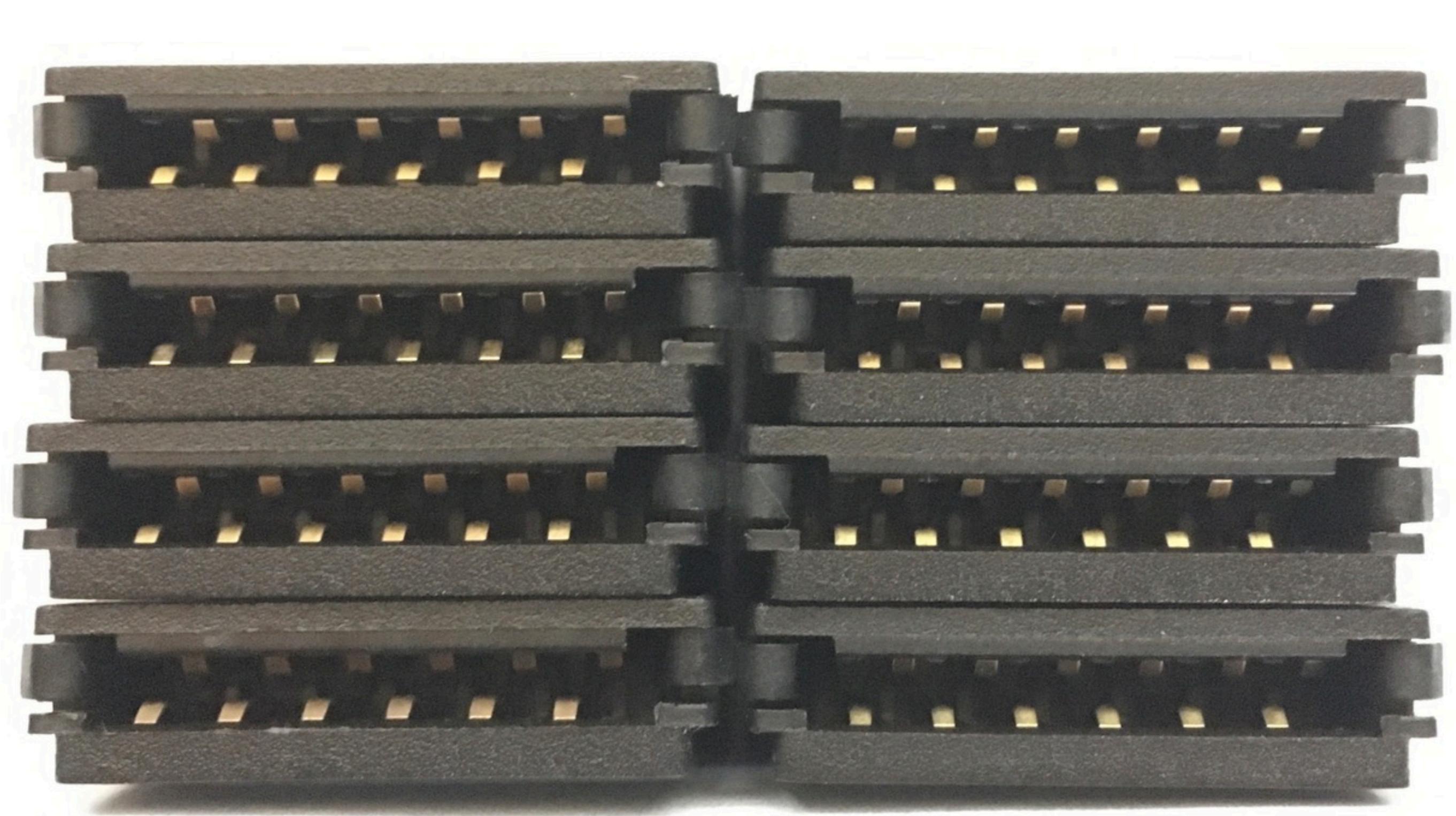
- In this presentation we have ...
 - Reviewed some key informations about the HP-41C system.
 - Discovered Diego Díaz modules.
 - Covered every options of three modules.
 - Gone through the programming process.
 - Configured NoV HEPAX emulation.
 - Cleared NoV HEPAX RAM content.

Questions & Answers



Index

| | | | |
|---|----|---|-----|
| Front Cover | 1 | Configuration | 39 |
| Agenda | 2 | Clonix-D Configuration | 40 |
| Introduction | 3 | Standard | 41 |
| Overview | 4 | Standard + Merged Blocks | 42 |
| Notes | 5 | Advantage | 43 |
| Acknowledgements | 6 | Forth 41 C/CV | 44 |
| Goals | 7 | Forth 41 CX | 45 |
| ference | 8 | Advantage + Forth 41 C/CV | 46 |
| Memory Types | 9 | Advantage + Forth 41 CX | 47 |
| HP-41C ROM Words | 10 | Dual X-Memory | 48 |
| HP-41C RAM Registers | 11 | Service 41C/CV | 49 |
| HP-41C ROM Memory Map | 12 | Service 41CX | 50 |
| EPROM ROM File Format | 13 | Service 41C/CX | 51 |
| ERAMCO ROM File Format | 14 | NoV-64(d) Configuration | 52 |
| HEPAX ROM File Format | 15 | HEPAX | 53 |
| Padded ROM File Format | 16 | HEPAX + Merged Blocks | 54 |
| MOD ROM Format | 17 | HEPAX + Advantage | 55 |
| MOD File Format | 18 | NoV's RAM Clear | 56 |
| MOD File Example | 19 | HEPAX RAM | 57 |
| LIF File Header | 20 | HP-41CY & RAMBOX64 | 58 |
| HEPAX 4K RAM Structure | 21 | Dual X-Memory | 59 |
| Overview | 22 | Service 41C/CV | 60 |
| Modules | 23 | Service 41CX | 61 |
| History | 24 | Quasi-ROM | 62 |
| Timeline | 25 | Clonix-D Persona | 63 |
| Clonix 41d | 26 | USB-41 Configuration | 64 |
| Specifications Table | 27 | HP-82143A | 65 |
| Specifications Notes | 28 | Programming | 66 |
| Resources and web links | 29 | Hardware 1 | 67 |
| Software | 30 | Hardware 2 | 68 |
| Clonix & NoV Configuration Utility v4.2 | 31 | Hardware 3 | 69 |
| Clonix & NoV Configuration Utility v6.1 | 32 | Software | 70 |
| MPASM | 33 | PIC Assembly & Intel Hex Files | 71 |
| MicroBurn DIY K150 | 34 | File Generation Failed | 72 |
| PICkit 2 Programmer | 35 | File Generation Successful | 73 |
| Device Manager | 36 | Programmer Not Found | 74 |
| USB 82143A | 37 | File Upload Failed 1 | 75 |
| USB-41 Page Transfer | 38 | File Upload Failed 2 | 76 |
| | | File Upload Successful | 77 |
| | | NoV Runtime Configuration | 78 |
| | | 16K RAM Mapping | 79 |
| | | 32K RAM Mapping | 80 |
| | | 64K RAM Mapping | 81 |
| | | 24K Flash Mapping | 82 |
| | | 48K Flash Mapping | 83 |
| | | Control Word | 84 |
| | | Control Word : NoV-32 | 85 |
| | | Control Word : NoV-64(d) | 86 |
| | | Crash Recovery Function : NoV-64(d) | 87 |
| | | ROM Shadowing & | |
| | | QROM Protection : NoV-64(d) | 88 |
| | | NoV Notes | 89 |
| | | HEPAX RAM Clearing | 90 |
| | | HEPAX vs NoV Modules | 91 |
| | | Clearing with Configuration Utility | 92 |
| | | Configuration | 93 |
| | | Clearing | 94 |
| | | Manual Clearing | 95 |
| | | Configuration | 96 |
| | | FOCAL Program | 97 |
| | | Clearing : NoVRAM & NoV-32 | 98 |
| | | Clearing : NoV-64(d) [16K] | 99 |
| | | Clearing : NoV-64(d) [32K] | 100 |
| | | Closing Topics | 101 |
| | | Review | 102 |
| | | Questions & Answers | 103 |
| | | Index | 104 |
| | | Back Cover | 105 |



DIEGO'S HP-41 MODULES