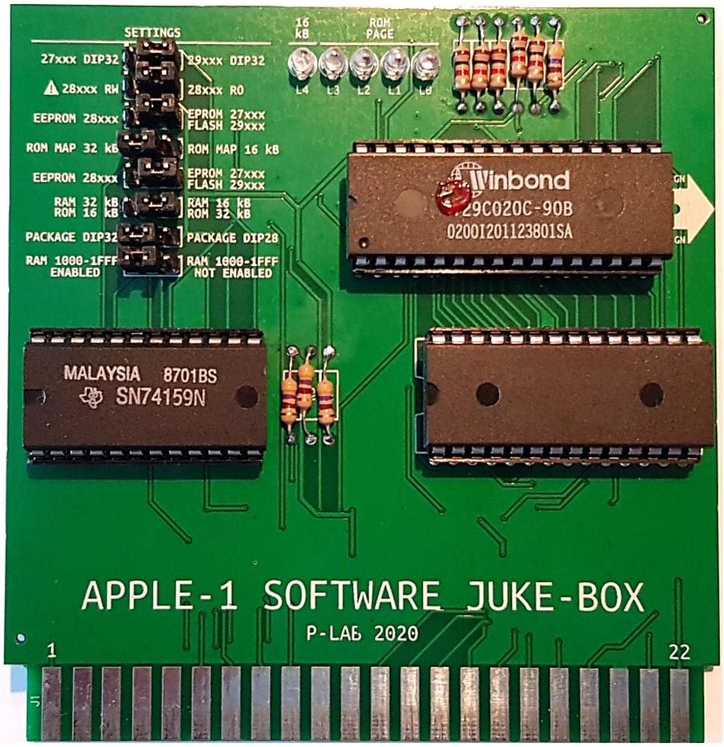
APPLE-1 JUKE-BOX

v1.09

Thank you for choosing Apple-1 Juke-Box, we hope it will meet your needs!
Please read this guide carefully and save it for future reference.

This document contains:

1. Package contents
2. Warnings about the risk of damage due to electrostatic discharges
3. Product description and features
4. Instructions for setting up and installing the Apple-1 Juke-Box
5. Instructions for loading programs
6. Configuration examples for different media



Picture above is for illustration purpose only. Actual product may vary due to due to product enhancement or availability of components.

1. PACKAGE CONTENTS

- *Apple-1 Juke-Box* board, with storage device such as 29c020, 27c020 or any other compatible device, tested and already set up.
- EEPROM 28c256 R/W, already programmed.

2. DAMAGES FROM ELECTROSTATIC DISCHARGES

Apple-1 Juke-Box is sensitive to static electricity, *just like your Apple computer*, and may be damaged by it.
Before any operation on your devices, you must discharge the static electricity accumulated by your body and prevent it from building up again. We do not accept any responsibility for damage, even serious or fatal, caused to people / things / intellectual property during the installation or use of this device.

3. PRODUCT DESCRIPTION AND FEATURES

Apple-1 Juke-Box is an electronic board that contains most of the software ever written for the **Apple-1 computer**.
Through a convenient and intuitive menu you can load programs and games chosen from those available, just like when you choose a song from a Juke-Box.
Thanks to the firmware stored on it, the card is able to load both programs written in BASIC and Machine Language.
The card is compatible with both Original and Replica computers.
In contrast to cassette tapes and programs recorded in audio format, **program loading is immediate**.
A separate document will explain how to save your programs permanently on a dedicated ROM.

The card is compatible with various storage media devices with capacities between 128 kbit (16 kByte) and 4 Mbit (512 kByte), as described below.

One of the peculiarities of the Apple-1 computer is to have, in its basic configuration, a not very populated memory map. This enables the allocation of large portions of addresses to external devices, such as the one you have just purchased.

The *Apple-1 Juke-Box* does not need any special jumper configurations in the Chip Select area of the computer.
As you will see below, the "with ACI" configuration is the recommended one.
Apple-1 Juke-Box therefore also works on the **original Apple-1 equipped with 8 kBytes or RAM, without any alteration**.

Let's see some of its features, which can be configured directly by you.

12 kByte RAM EXPANSION: in this configuration the total contiguous RAM of the computer is 16 kByte, (from 0x0000 to 0x3FFF), plus the 4 kByte usually reserved to BASIC (from 0xE000 to 0xEFFF), for a total of **20 kByte RAM**.

28 kByte RAM EXPANSION: in this configuration the total contiguous RAM of the computer is 32 kByte (from 0x0000 a 0x7FFF), plus the 4 kByte usually reserved to BASIC (from 0xE000 a 0xEFFF), for a total of **di 36 kByte RAM**.

PERMANENT ARCHIVE OF BASIC and Machine Language PROGRAMS or GAMES. The card can allocate up to 4 Mbit of memory (512 kBytes), each “page” of 16 or 32 kBytes can be addressed separately. A single page can contain several programs. Through the configuration jumpers, the card is compatible with the majority ROM/PROM/EPROM/EEPROM/FLASH/OTP available on the market, with a capacity between 16 kBytes and 512 kBytes. In this document we will always refer to these devices by the generic term "ROM".

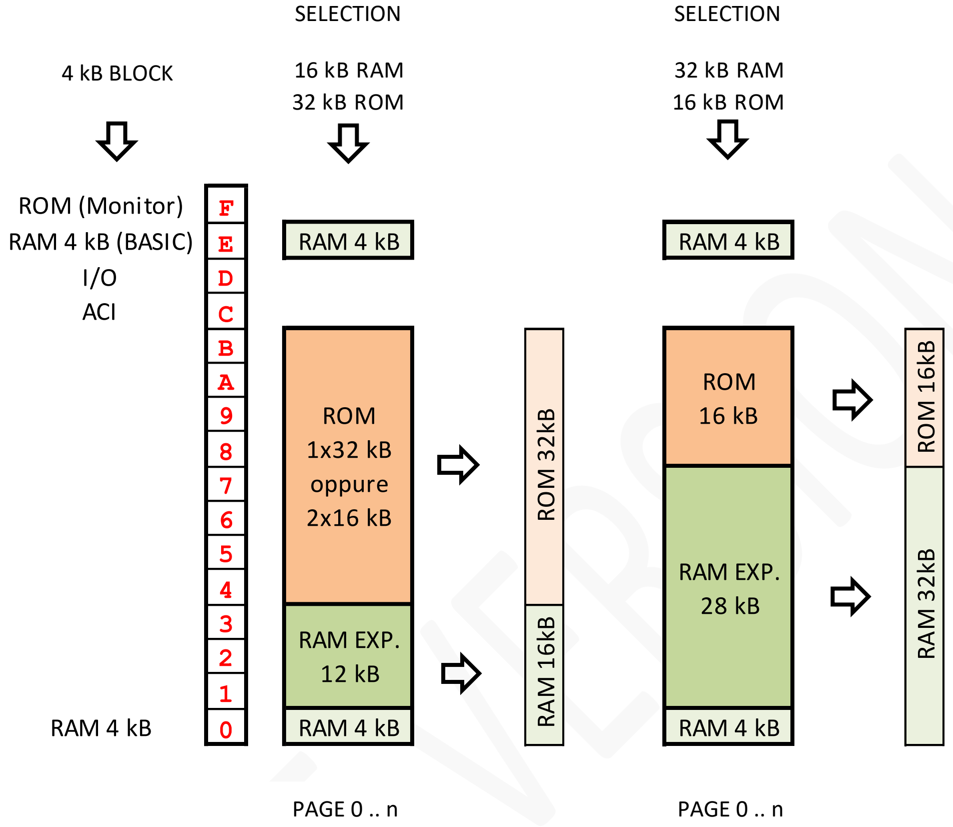
The "logical" dimension of each ROM page can be chosen between:

- **16 kByte per page** (address space between 0x8000 and 0xBFFF): to choose whether the programs you want to run need a lot of RAM or whether they need to be allocated on memory portions higher than 0x3FFF. This is the case, for example, of certain) extended BASIC interpreters, or other programming languages, etc.).
- **32 kByte per page** (address space between 0x4000 and 0xBFFF): to choose if the programs to be run do not require more than 16 kBytes of RAM. This is the most frequent case and usually applies to the majority of programs and games developed in BASIC.

A 32 kByte ROM can therefore be organised during preparation/writing as a *single 32 kByte page* or as *two 16 kByte pages*. Similarly, a total 64 kByte ROM can be organised as *two 32 kByte pages* or *four 16 kByte pages*.

A hybrid configuration is also possible. With the above example: a 64 KByte ROM can be split into one 32 kByte page plus two 16 kByte pages.
In this case, however, it will not be possible to access the second 16 kByte page without first switching off the computer and changing the configuration jumpers.

The following drawing shows the allocation of the card in the computer memory map.



The two configurations indicated above are "alternative" and must be appropriately set up before the computer is switched on, depending on the type of program you intend to run.

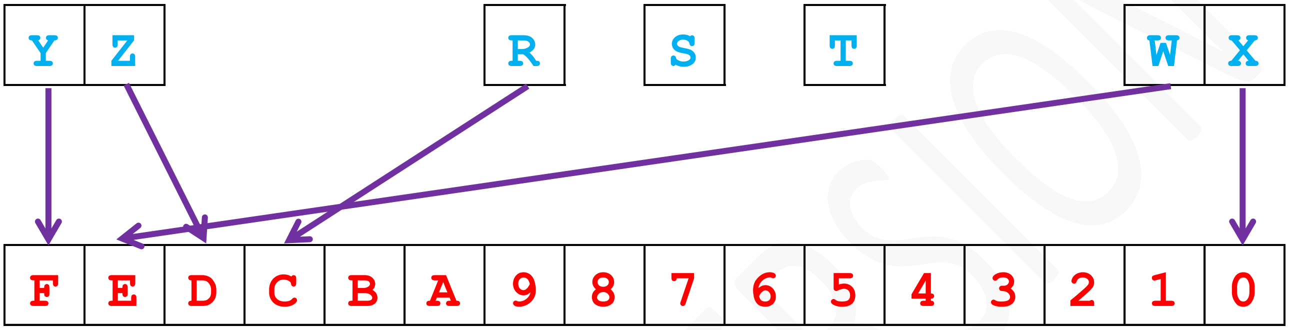
As said, the configuration on the left, with more ROM, will be the one usually used to run most of the BASIC programs. In nearly all cases, in fact, they will never need more than 16 kBytes of contiguous RAM.

The use of 16 kByte logic mapping will be discussed in the forthcoming sections.

4. SET-UP AND INSTALLATION

4.1 INITIAL CHECKS

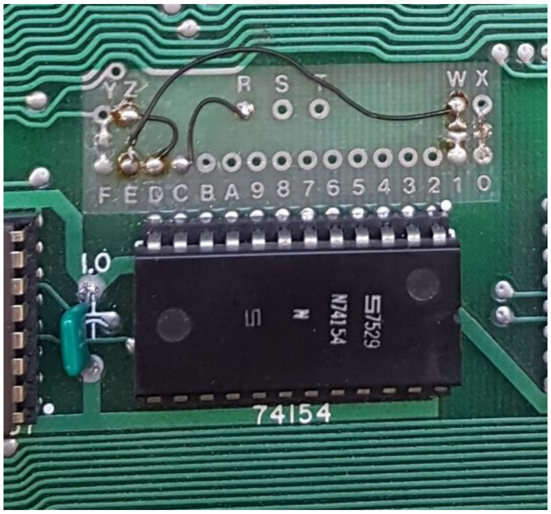
Apple-1 must be set up with the configuration “WITH ACI”, with the following connections in the CHIP SELECT area:



Jumper “1000-1FFF RAM” must be placed on the <u>ENABLED</u> side.

NOT ENABLED position should only be used for testing.

Picture of the necessary connections/wires:



4.2 SETTING THE JUMPERS ON THE CARD

Given the number and diversity (also physical) of ROM formats supported, the card must be set up accordingly to the model used, so that all electrical connections are made in the correct way.

An incorrect configuration could lead to malfunctions and damage to both the card and the computer.

At the moment the card has been tested with the following devices:

- EPROM 27c128 / 256 / 512 (DIP package 28) - EPROM 27c020 (DIP package 32)
- EEPROM 28c256 (DIP 28 package)
- FLASH 29c020 / 29c040 (DIP 32 package)

Other models might be compatible.

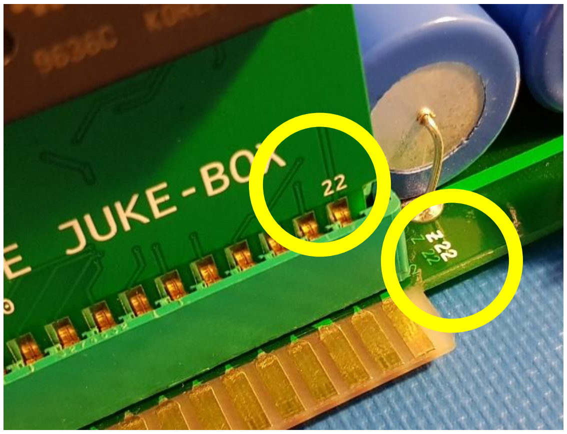
The various jumpers and their configuration criteria are listed below.
Always check them all when changing ROM types.

JUMPER	FUNCTION
PACKAGE DIP32 / DIP28	Place the jumper on the side corresponding to the number of pins of the ROM in use: 28 (14 per side), or 32 (16 per side). Be sure to position the component by aligning it with the right edge of the board, in the direction indicated by the arrow. In case of PACKAGE type DIP-28 the LED L3 will be permanently lit.
RAM 32 kB-ROM 16 kB / RAM 16 kB-ROM 32 kB	This jumper implements the PHYSICAL mapping of RAM and ROM already illustrated in the "features" paragraph.
EEPROM 28xxx / EPROM 27xxx - FLASH 29xxx	Two Jumpers with the same indication. Place BOTH on the side corresponding to the device in use.
27xxx DIP 32 / 29xxx DIP 32	If you are using a device with a DIP 32 package, place the jumper in the appropriate side. This jumper is ineffective if you are using a ROM with DIP 28 package.
28xxx RO / 28xxx RW	In RW position it enables writing on EEPROM 28c256 (see dedicated document), in RO position the EEPROM will be in read-only mode.
ROM MAP 32 kB / ROM MAP 16 kB	Indica al programma di gestione che la mappatura LOGICA della ROM è su 16 kB o 32 kB (vedi paragrafo dedicato).

NEVER CHANGE THE POSITION OF THESE JUMPERS WITH THE COMPUTER SWITCHED ON

4.3 INSTALLATION

The card must be inserted as shown in the picture:



The number "22" on the card and the number "22" on the motherboard must be on the same side.
The number "22" on the card must therefore face the outer side of the Apple-1 motherboard.

Using the card in combination with other cards and/or devices such as BUS EXTENDER could lead to malfunctions: do not use this kind of devices. The Juke-Box card must be the only one present.

WARNING: Switching on the computer with the card oriented incorrectly **INSTANTLY DAMAGES** the computer and the card itself.

5 PROGRAM LOADING INSTRUCTIONS

The card has its own program manager which allows you to easily load, at your choice, the BASIC interpreter, programs written in BASIC and programs written in Machine Language.
First turn on the computer, reset it and clear the screen as usual.

5.1 THE PROGRAM MANAGER

To run the Program Manager type in:

`BD00R` and press ENTER (from now on indicated with {ENTER})

The following will be displayed:

```
BD00: A5
&
```

The "&" symbol is the prompt generated by the Program Manager. It indicates that it is ready to accept commands.

5.2 HELP SCREEN

A little help is available, it can be recalled with the **H** command:

```
&H  {ENTER}
D)IR            shows files and programs in the current page
L)OAD ID        load file/program corresponding to the displayed letter
S)ET 0.1        set upper/lower 16 kByte in case of 16 kByte logic mapping
P)AGE 0.F       select one among the 16  (0, 1, 2… C, D, E, F) 32 kByte pages
B)ASIC          enter BASIC with E2B3R (non-destructive entry-point for BASIC)
E(X)IT          exit and go back WOZ monitor
&
```

Each command is recalled with its own initial, except the EXIT command.

5.3 PAGE SELECT

The page selection is performed with the **P command followed by the desired page number**, in single digit hexadecimal format. This means that 16 pages (from 0 to F included) can be addressed, each of which is 32 kByte wide. This brings the maximum manageable memory size to 512 kByte, more than enough to accommodate a large number of programs.
To select, for example, page 2 the command will be:

```
&P2  {ENTER}
 OK             The Program Manager replies with OK. The LEDs below the PAGE indication will light up
                accordingly to the binary value of the requested page. This is a useful indication.
&               Program Manager is ready for the next command.
```

If the ROM has been logically mapped in 16 kBytes: refer to section 5.7.

5.4 FILE LISTING

To retrieve the list of available files: use the **D** (Directory) command:

```
&D  {ENTER}
 PAGE 2

 A CHEMIST  $0C33-$1000 BAS
 B DEMO MLR $0280-$11F8
 C BASIC    $E000-$F000
 D LITTLETR $0300-$14BD
 E HANGMAN  $0DA9-$2000 BAS
 F LUNARLN2 $0300-$19B9
 G CODEBRKR $0280-$1AB7
 OK

&
```

The current page (PAGE) is shown, and all files present are then listed.
Each file is identified by a letter **from A onwards**, for a total of a maximum of 17 programs per page.

Each file has a name (maximum eight characters), followed by the starting memory address and the first free memory address.

If "BAS" appears, this means that the program has been written in BASIC. The BASIC interpreter must be loaded before the program marked "BAS" is run.

5.5 LOADING BASIC PROGRAMS

The BASIC interpreter is a program like any other, so it will be loaded with the **L** command followed by the letter assigned to it by the Program Manager. The following line appears in the previous example:

```
 C BASIC    $E000-$F000
```

In order to load the BASIC interprer give the following command:

```
&LC  {ENTER}
 OK
```

The Program Manager replies immediately with OK indicating that the "BASIC" program has been loaded into the memory.

At this point you can load the desired program.
Suppose you want to play HANGMAN. In the file directory it is identified with the line:

```
 E HANGMAN  $0DA9-$2000 BAS
```

The command to give now is therefore:

```
&LE  {ENTER}
 OK
```

The Program Manager replies immediately with OK indicating that the "HANGMAN" program has been loaded into the memory.

At this point both the program and the BASIC interpreter are loaded onto the memory, there is nothing left to do but enter the BASIC environment (in a non-destructive way for the program just loaded) with the appropriate **B** command:

```
&B  {ENTER}
>
```

The ">" prompt indicates that you are in the BASIC environment, as you know. To run the newly loaded program, simply give the usual RUN command, and the game will start.

```
>RUN  {ENTER}
                HANGMAN
CREATIVE COMPUTING MORRISTOWN, NEW JERSE
Y

HERE ARE THE LETTERS YOU USED:

----
WHAT IS YOUR GUESS?
```

5.6 LOADING OTHER PROGRAMS

The procedure for loading programs written in Machine Language is absolutely identical to the one used to load BASIC (which is also a program written in Language).
In our example page we have already met LITTLE TOWER:

```
D LITTLETR $0300-$14BD
```

After entering (or re-entering, if necessary) the program manager with the BD00R command, check with the D and P2 commands that you are in the correct ROM page. To load the program, simply give the command:

```
&LD
 OK
```

{ENTER}

The Program Manager replies immediately with OK indicating that the "LITTLETR" program has been loaded into the memory.

```
&X
```

{ENTER} It is now necessary to return to the WOZ Monitor, since it is not a BASIC program.

Then it is necessary to run the newly loaded program. To do this, it is essential to know the hexadecimal *entry-point*, which is information supplied with the program itself.
Usually, but not always, it corresponds with the first address used by the program.
In the case of the Little Tower program it is indeed 0x0300. The following command must therefore be given:

```
0300R
```

{ENTER}

The game will start immediately:

```
0300: D8

      LITTLE TOWER
        APPLE-1 COOL GAME
          WRITTEN BY A.VERHILLE
              ...
```

...and now you can start a new game.

PLEASE NOTE. In the games and programs provided the hexadecimal *entry-point* (if different from the first address occupied by the program) has been written in the name itself. Example:

```
J WOZF2394 $1D00-$2B34
```

In this example the program *Woz's floating point demo* has as entry-point 0x2394, not 0x1D00.
So, if the last four characters in the title of a "non BASIC" program are an hexadecimal address, use those as *entry-point:*

```
2394R
```

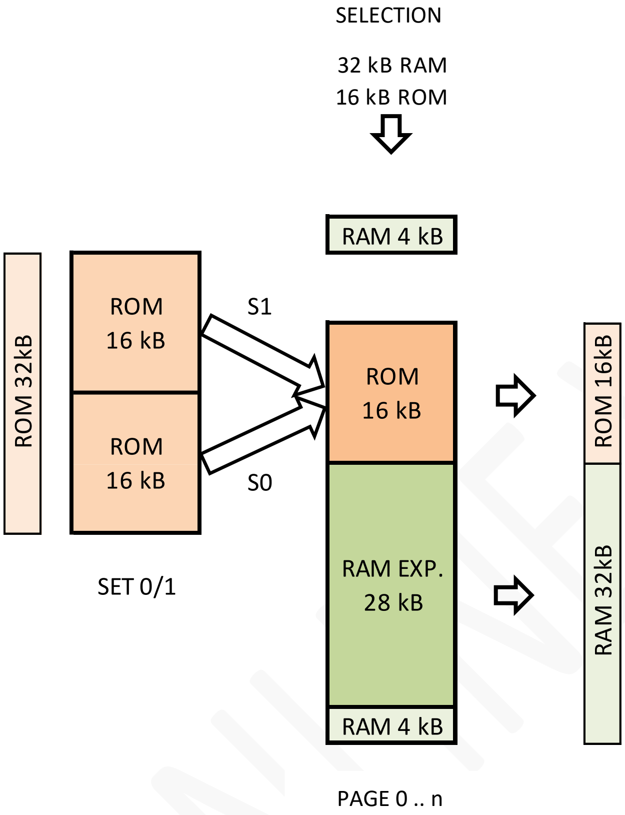
{ENTER}

5.7 16 kByte MAPPED ROM

A 32 kByte (or more) ROM can be divided into 16 kByte portions. This option is particularly useful, if not indispensable, when the programs to be run need to be allocated to a memory segment greater than 0x3FFF.
This is the case, for example, of alternative languages to the classic INTEGER BASIC, or other utilities.
In this case, the ROM must be prepared in advance in an appropriate format.

In order to correctly access the second 16 kByte LOGIC portion within a 32 kByte PHYSICAL ROM, the S (SET) command has been prepared.

This command, followed by 0 or 1, addresses the corresponding 16 kByte sub-page.



The picture shows the operation of the SET control.

The S1 command causes the "16 k" LED on the board to turn on and the upper 16 kByte part to be accessible.

The S0 command switches it off and makes the lower 16 kByte part accessible.
This is the default condition.

5.8 IMPORTANT INFORMATIONS

- To load programs and games from *the Apple-1 Juke-Box*, you must **always start the Program Manager first** with the BD00R command from the WOZ Monitor.
- Usually it is necessary to load the **BASIC interpreter only once**: if you decide to change the program/game, simply load the new program and re-enter the BASIC environment with the B command, and then RUN.
- BASIC interpreter and programs **may be on different pages**. You can then load BASIC from page 2, for example, and your favourite game from page 5.
- Depending on the capacity and mapping of the ROM, the pages may be "duplicated". This is perfectly normal and not a malfunction. For example a Flash memory 29c020 (2 Mbit, 256 kByte, 8x32 kByte pages will display the same contents on page 0 + page 8, page 1 + page 9, page 2 + page A and so on.
- If no BASIC program has been loaded, the command **B issued from the Program Manager will not work and will hang the computer** (in case: just RESET). This happens because the pointers needed by BASIC interpreter are loaded with the program itself: No BASIC program → No BASIC. If you want to enter the BASIC environment and start writing your own program, you must load the BASIC interpreter, exit the WOZ Monitor with the X command and **initialise the BASIC environment from scratch** with the E000R command. Keep in mind that E000R erase any BASIC program.
- Command B of the Program Manager is equivalent to command X followed by command E2B3R (soft entry point of BASIC, non-destructive for previously loaded programs).
- Normally all programs in RAM survive the Hardware RESET.
- Entering **wrong or not allowed** commands will cause a "!" symbol to appear.
- Attempting to load a non-existent program (e.g. program with index "N" when the current page reaches only index "F") will have no effect and **will not generate any error message**.

5.9 NOTE FOR TEXAS INSTRUMENTS EPROMS

Some EPROMs, in particular the Texas Instruments TMS 27c128, may be perfectly written by your programmer but totally unreadable by the Apple-1 Juke-Box.
This behaviour is due to a particular evaluation of a specific control signal by the EPROM itself.

In this circumstance it will be impossible to launch the management program with the BD00R command.
To solve this inconvenience, write the value 0x11 in locations 0x003F and 0xCA00 with the following commands:

```
003F:11  {ENTER}
CA00:11  {ENTER}
```

Run the Program Manager with the usual BD00R command and continue normally.

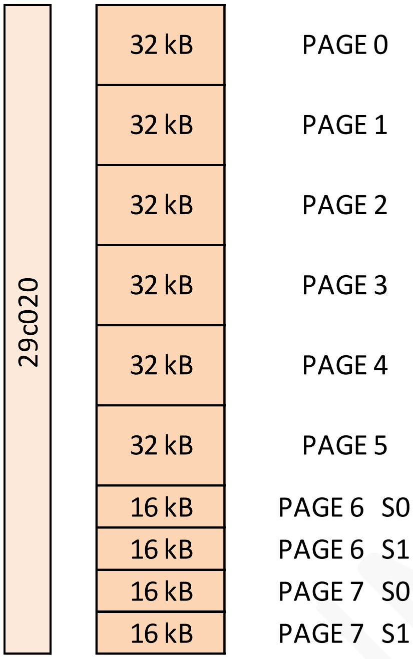
6 CONFIGURATION EXAMPLES

The card comes with two storage media:

- 29c020 -- FLASH Memory, 256 kByte or compatible EPROM such as 27c020 or similar, with same capacity.
- 28c256 -- EEPROM Memory, 32 kByte

The required settings for each of the supports are shown below.

6.1 29c020



This medium has been designed to have eight pages: six 32 kBytes mapped on 32 kBytes and two pages mapped as 2x16 kBytes.
If you want to access pages 6 and 7 correctly, the configurations of the jumpers must therefore be changed as indicated, with the computer switched off.

JUMPER	SETTING FOR 29c020
PACKAGE DIP32 PACKAGE DIP28	PACKAGE DIP32
RAM 32 kB-ROM 16 kB RAM 16 kB-ROM 32 kB	RAM 32 kB-ROM 16kB for pages from 0 to 5 RAM 16 kB-ROM 32kB for pages 6 and 7
EEPROM 28xxx EPROM 27xxx - FLASH 29xxx	EPROM 27xxx - FLASH 29xxx
27xxx DIP 32 29xxx DIP 32	29xxx DIP 32
28xxx RO 28xxx RW	INDIFFFERENT
ROM MAP 32 kB ROM MAP 16 kB	ROM MAP 32 kB for pages from 0 to 5 ROM MAP 16 kB for pages 6 and 7

Page 6 and 7 contain a total of four 16 kByte pages hosting alternative programming languages. They require more RAM memory than is usually used by the INTEGER BASIC and its programs. To use these programs, set the jumpers as indicated above "for pages 6 and 7". Then select the desired sub-page with the S0 and S1 commands.

Please note:
If the board come with 27c020 storage device, the jumper 27xxx DIP 32 /29xxx DIP 32 has to be placed on the 27xxx side.
The "Adventure" game only, stored on page 0, requires a special configuration.
It requires 32 kBytes of RAM: if loaded with the above-mentioned jumpers configured for pages "0 to 5", it will not work. To load and run it successfully, it is necessary to set the jumpers as indicated for pages "6 and 7".
Then load it normally and run it with **0280R**.

6.2 28c256

28c256

32 kByte

PAGE 0

This medium has been logically mapped as a single 32 kByte page. It allows you to write BASIC programs and save them permanently. For more details see the dedicated document.

JUMPER	SETTING FOR 28c256
PACKAGE DIP32 PACKAGE DIP28	PACKAGE DIP28
RAM 32 kB-ROM 16 kB RAM 16 kB-ROM 32 kB	RAM 16 kB-ROM 32kB
EEPROM 28xxx EPROM 27xxx - FLASH 29xxx	EPROM 28xxx
27xxx DIP 32 29xxx DIP 32	INDIFFERENT
28xxx RO 28xxx RW	Choose: RO for read only, RW for read/write
ROM MAP 32 kB ROM MAP 16 kB	ROM MAP 32 kB

We hope you enjoy using *Apple-1 Juke-Box!*

APPLE-1 JUKE-BOX

INFO | RFQ | SUPPORT: p-l4b @ protonmail.com

P-L4B @ PROTONMAIL.COM