Garrett’s Workshop

GW4203B “RAM2E II”

RAM Expansion Card for Apple IIe

User’s Guide

# Overview

RAM2E II (GW4203B) provides the Apple IIe with 8 MB of auxiliary memory and enables double-high-resolution graphics and 80-column text display modes.

## Low-Power, SDRAM-Based Design

Thanks to a modern, low-power design, RAM2E II uses only 0.2 watts (40 mA @ 5V) in typical use and 0.4 watts (80 mA @ 5V) in heavy use with an accelerator[[1]](#footnote-1). In contrast, a typical 80-column card consumes over 1 watt of power, and a 1 MB RAMWorks III consumes 2 watts or more. Unlike other IIe auxiliary memory cards, which are built with vintage asynchronous DRAM chips, RAM2E II uses modern SDRAM. This design allows for low power consumption and improved reliability over other memory cards using 15+ year old chips.

## Small Size, Low-Profile

RAM2E II features a small board outline and is the thinnest Apple IIe auxiliary memory card ever produced, at under 4mm thin. Small and thin dimensions improve the mechanical compatibility between RAM2E II and peripheral cards installed into the IIe's Slot 1.

## Adjustable Capacity, Highly Compatible

RAM2E II is highly compatible with existing software that utilizes RAMWorks memory standard. An adjustable capacity feature allows the memory size to be set to 64 kB, 512 kB, 1 MB, 4 MB, or 8 MB, improving compatibility with software expecting a particular memory capacity. Capacity settings can be set temporarily or saved in nonvolatile memory. The capacity adjustment utility is available on our website: <http://garrettsworkshop.com>

## Ecologically Friendly, Gold-Plated PCB

RAM2E II is built with a lead-free, ENIG gold-plated, 4-layer PCB and is fully EU RoHS-compliant. All units are tested extensively before shipment, and only new parts are used to build RAM2E II.

## Open-Source Design

RAM2E II's design is fully open-source. The schematics, board layouts, CPLD firmware, and utility software are all freely available for commercial and noncommercial use. To download the design files, visit the Garrett's Workshop GitHub page: <https://github.com/garrettsworkshop>

## Note for Revision A Apple IIe Owners

If you own the uncommon revision A Apple IIe and are interested in RAM2E II, please read this! RAM2E II requires a small modification to be compatible with the with the rev. A Apple IIe. The modification consists of removal of the “DHGR” jumper resistor or cutting the associated jumper trace.

Revision A machines can be identified by their part number “820-0064-A” printed at the top of the motherboard near the slots. If you have a rev. A Apple IIe, please contact us before purchasing and we can make the modification before shipping your RAM2E unit. However, do note that with the modification made, RAM2E II will not support the double-high-resolution graphics display mode.

## Note for A2Heaven VGA Scaler Owners

If you own the A2Heaven VGA Scaler and are interested in RAM2E II, please read this! There is a slight incompatibility between RAM2E II and the A2Heaven VGA Scaler as shipped.

As owners of the VGA Scaler will know, a long jumper wire is required to connect the VGA Scaler to the Apple IIe’s 14 MHz clock signal. This arrangement causes glitches in the 14 MHz clock signal and is therefore not compatible with RAM2E II. The solution to the problem is to use a special cable to connect the VGA Scaler to the 14 MHz clock in place of the cable supplied with the VGA Scaler. Please let us know if you plan to use RAM2E II with the VGA Scaler and we can include the requisite hardware to address the problem.

A more technical description of the issue with the A2Heaven VGA Scaler follows: The incompatibility involves the mishandling of the Apple II’s 14.318 MHz master clock signal by the VGA Scaler. The VGA Scaler connects to the 14.318 MHz clock through a long wire. The addition to the clock signal line of this wire in combination with the input capacitance of the VGA Scaler’s FPGA causes significant distortion to the clock waveform. This distortion includes ringing and long rise times. While the slower response time of the ICs in the Apple IIe makes the Apple itself insensitive to this distortion of the clock signal, a more modern card such as RAM2E II is susceptible to these clock signal artifacts. In our testing, the jumper wire arrangement can cause instability and periodic memory errors in RAM2E II. Adding an additional series resistance to the clock wire going to the VGA Scaler minimizes the impact of the wire arrangement and allows the system to run stably.

# Installation

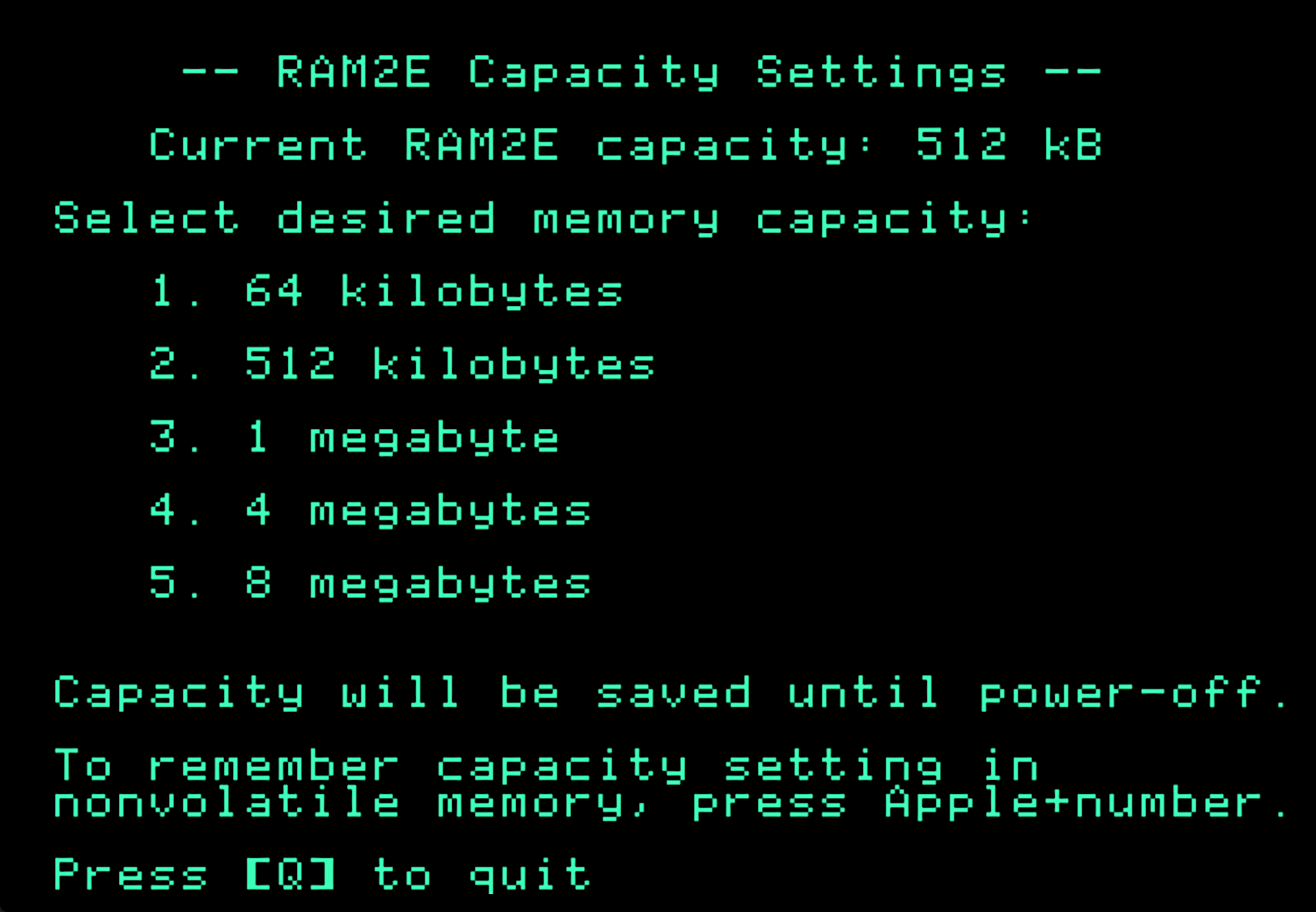
RAM2E II must be installed into the Apple IIe auxiliary slot. Do not install the card into one of the seven expansion card slots. Also ensure that RAM2E II is inserted in the correct orientation. Markings on the card indicate the side which must face towards the power supply of the Apple IIe.

# Changing RAM Capacity

In order to maximize compatibility, the memory capacity of RAM2E II can be changed to 64 kilobytes, 512 kilobytes, 1 megabyte, 4 megabytes, or 8 megabytes. The capacity can be set either temporarily until the Apple IIe is powered off, or in nonvolatile memory where the setting will persist.

Capacity adjustment is accomplished using the “GWRAM” utility program available for download from our website at <http://garrettsworkshop.com/files/GWRAM/GWRAM.po>. The program is packaged as a 140 kB .po format disk image containing the GWRAM utility and ProDOS 2.4.2.

After launching GWRAM on an Apple IIe with a RAM2E II card, the following menu is shown:



For example, by pressing the “1” key, the capacity can be set to 64 kilobytes until the computer is powered off. By pressing Apple+1 (open-Apple or closed-Apple), the 64 kilobyte capacity setting can be saved in nonvolatile memory as well, so that it is restored on power-up.

# Technical Specifications

## Physical Dimensions

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| **Height** |  |
| **Width** |  |
| **Thickness** |  |
| **Weight** |  |

## Electrical Specifications

Specifications are valid over temperature range of and .

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Value** | **Conditions** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| **Output Slew Rate** |  |  |
|  |  |  |
|  |  | data bus MD[7:0], C14M clock signal |
|  | all other signals |
|  |  |  |

# Information for Developers: RAM2E II Command Set

RAM2E II supports a proprietary command set to facilitate software access to the special features of the card (for example to adjust the RAM capacity). The content of a command consists of an 8-bit command number and an 8-bit argument. Commands are issued by repeated writes to the RAMWorks bank register at $C073. To issue a command, a series of eight write accesses to the RAMWorks bank register at address $C073 is required. Each of the eight bytes must be written within seven PHI0 clocks of the submission of the previous byte or the command will not be accepted. This prevents ordinary RAMWorks bank switches from inadvertently triggering a command submission. In the first part of the command sequence, six magic numbers must be written in sequence to $C073. The magic number sequence which must be written is $FF, $00, $55, $AA, $C1, $AD. Following writing the magic number sequence to $C073, the command number and then argument are written to $C073. Once all eight bytes are submitted, the command is executed. If eight or more PHI0 cycles elapse between submission of subsequent bytes of the command sequence, or if an incorrect magic number is submitted, the command sequence detector resets and the command sequence must begin again for any command to be accepted. After a command is submitted, unless otherwise specified in the command description, the RAMWorks bank address will be equal to the argument submitted, as the argument is the last byte written to $C073 as part of the command sequence. For more information on this command set, see the RAM2E and GWRAM utility program source code on the Garrett’s Workshop GitHub page at <http://github.com/garrettsworkshop>.

Sample code in 6502 assembly language is given below to submit a command to the RAM2E II. The code assumes that the command number is stored in the X register and the argument is stored in the Y register.

; Routine to submit a RAM2E II command

; Assumes command is in X register and argument is in Y register

; First reset command sequence just in case

lda #$00

sta $C073

sta $C073

; Send magic numbers to start command (FF 00 55 AA C1 AD)

lda #$FF

sta $C073

lda #$00

sta $C073

lda #$55

sta $C073

lda #$AA

sta $C073

lda #$C1

sta $C073

lda #$AD

sta $C073

; Send command and then argument (in X and Y registers respectively)

stx $C073

sty $C073

A table of supported commands is given below:

|  |  |  |
| --- | --- | --- |
| **Cmd. Name** | **Num.** | **Description** |
| **NOP** | $00 | No special meaning: RAMWorks bank is set to argument after command is executed. |
| **RWMaskSet** | $E0 | Sets the AND mask applied to the RAMWorks bank, thus setting the RAMWorks capacity. For example, a mask argument of $00 corresponds to a capacity of 64 kB, whereas $7F corresponds to a capacity of 8 MB. |
| **CFGBitbang** | $EA | Shifts configuration bit into holding register in preparation to write to CFG flash. Bits 0 through 5 of the argument must be 0. Bit 6 of the argument is the configuration bit to shift in. Bit 7 of the argument must be 1. Future versions of RAM2E II may not support the CFGBitbang command. |
| **CFGReset** | $EE | Causes the RAM2E II card’s CFG flash to be completely erased. This command is intended to repair the CFG flash if it is put in an invalid state. Avoid frequent use of the CFGReset command, as it causes wear-leveling information in the CFG flash to be erased, thus significantly decreasing the CFG flash’s remaining lifetime. This command can only be executed once per power cycle. Subsequent invocations of the CFGReset command are equivalent to a NOP command, preventing an errant program from writing to flash repeatedly and needlessly reducing the lifetime of the CFG flash. Argument must equal $00. Future versions of RAM2E II may not support the CFGReset command. |
| **CFGPrgm** | $EF | Causes the configuration data loaded using the CFGBitbang command to be written to CFG flash. This command can only be executed once per power cycle. Subsequent invocations of the CfgPrgm command are equivalent to a NOP, preventing an errant program from writing to flash repeatedly and needlessly reducing the lifetime of the CFG flash. Argument must equal $00. Future versions of RAM2E II may not support the CFGPrgm command. |
| **SetRWBankFF** | $FF | Sets the RAMWorks bank to $FF after command is executed, independent of argument value. Other auxiliary RAM cards will set their bank to the argument provided, rather than $FF. This allows software to detect RAM2E II. |

# Information for Developers: Theory of Operation

The operation of RAM2E II is somewhat different from that of other expansion RAM cards for the Apple IIe. While other cards are implemented with asynchronous DRAM chips, RAM2E II uses modern synchronous DRAM (SDRAM).

Since the IIe’s memory expansion slot was designed for use with asynchronous DRAM, additional circuitry on the RAM2E II card is required to interface the IIe with SDRAM. There are numerous differences between asynchronous DRAM and SDRAM, but the most significant is that in SDRAM, operation is pipelined over multiple clock cycles. Although SDRAM is a much newer technology than asynchronous DRAM, and therefore much faster, multiple clock cycle “steps” are required to perform an SDRAM access.

RAM2E II’s logic circuitry is implemented in a single CPLD which runs from the Apple IIe’s C14M 14.31818 MHz master clock signal. A buffered copy of this clock is supplied to both the RAM2E II’s CPLD and its SDRAM. Running from the Apple’s master clock, the RAM2E II translates auxiliary memory and 80-column video access commands issued by the Apple II into SDRAM commands which implement the same functionality.

The RAM2E II state machine runs from the C14M master clock of the Apple IIe. A 4-bit state counter is reset to 0x1 at the beginning of each PHI1 period and counts up to 0xE in a 14-clock cycle. In the last two clocks of a 16-clock “long cycle,” the state counter is equals 0xF. State 0x0 is only used during initialization.

To implement the auxiliary memory card functionality, SDRAM read/write commands are issued based on the current state index, the /WE80 signal, and the /EN80 signal. Unlike a traditional auxiliary RAM card, the /RAS and Q3 signals are not used for SDRAM control at all. The SDRAM command and address signals are implemented as registered outputs.

The data bus, as input to the RAM2E II and output to the SDRAM, is implemented as an asynchronous tristate buffer, and the data bus output is implemented as a registered output latched at the falling edge in the middle of state 0xC. Similarly, the video data bus output is registered at the falling edge of state 0x6. Both the video and 6502 data buses are output using 74AHCT-series buffers running at 5V. 74AHCT was chosen for its low power consumption, fast propagation delay, and slow output edge rate. Moreover, the 74AHCT-series outputs are desirable for their 3.8 V Voh specification at 8 mA of source current and 4.5V Vcc. This allows RAM2E II to meet the Vih specification of newer 65C02 processors with “pure CMOS” input buffers.

RAM2E II also supports a proprietary command set which allows software to adjust the RAM capacity and access other features.

## Information for Developers: Timing Diagram

The timing diagram given below shows the behavior of the major signals in the system, including the SDRAM command sequence used when reading and writing auxiliary RAM.A screenshot of a cell phone

Description automatically generated

1. Maximum power consumption measured with quadruple-unrolled auxiliary RAM read/write loop running on FastCHIP IIe accelerator at 16.6 MHz. [↑](#footnote-ref-1)