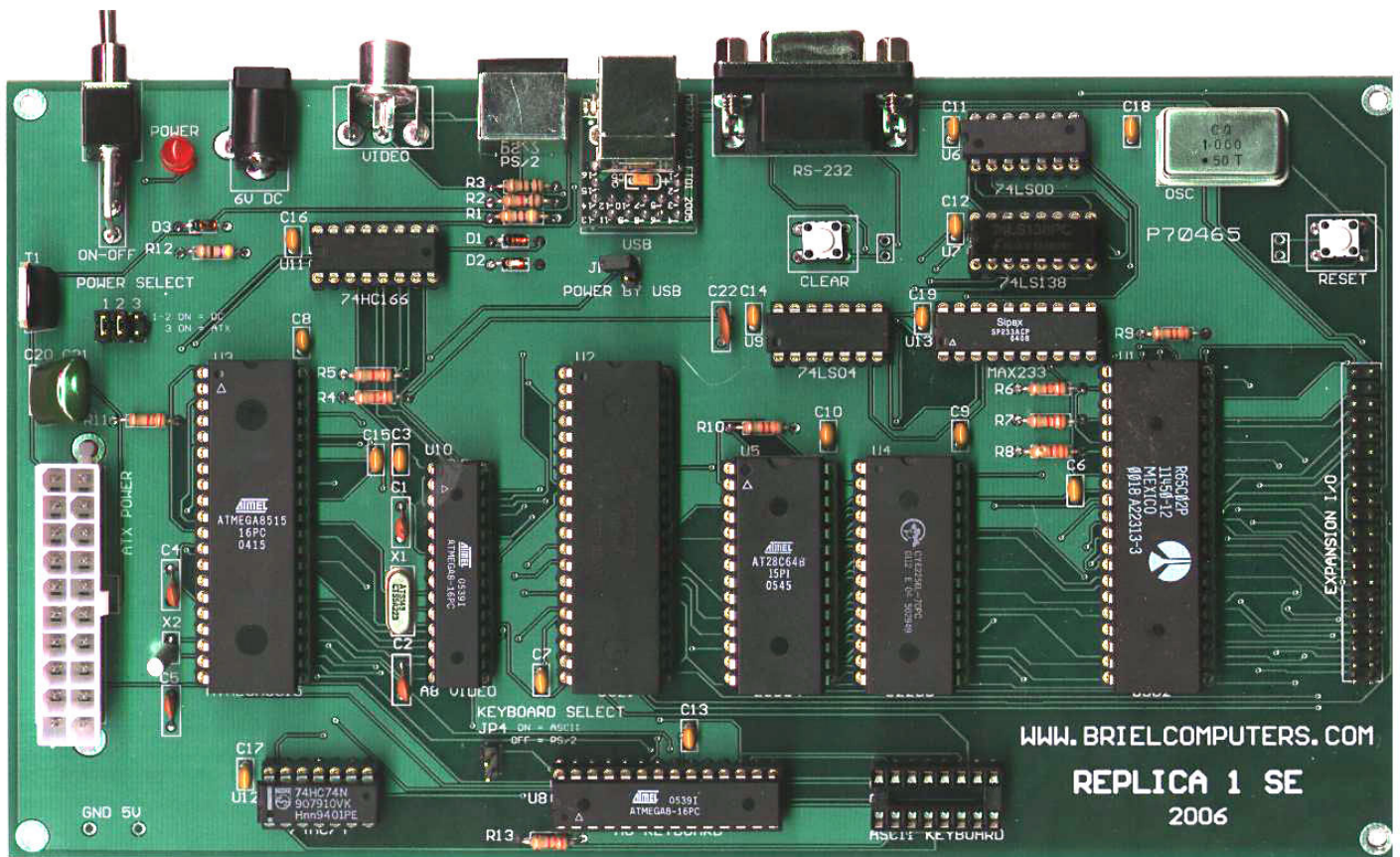


replica I

Setup and Users Manual

JANUARY 2007



Briel Computers
5392 Cornell Blvd
North Ridgeville, OH 44039

All materials, schematics, and hardware designs provided without any warranties. Although this material has been carefully examined, Briel Computers takes no responsibility for any errors in printing.

Version 2.0

Published by Briel Computers
5392 Cornell Blvd
North Ridgeville, OH 44039
USA

Copyright 2007 Briel Computers. All rights reserved. Printed in the United States of America. Except as permitted under the Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher, with the exception that the program listings may be entered, stored, and executed in a computer system, but they may not be reproduced for publication.

Forward

How the replica 1 came to be

Thank you for purchasing your replica 1 computer. I take pride in every kit and assembled replica that is shipped and my goal is that all kits and assembled boards work as designed. In early 2001 I was an avid collector of vintage computers. I had started with an Apple IIe because it was close to my first computer the Apple IIplus. I, like many people who get the urge to relive the yesteryears of computing looked to eBay to find my past. After receiving my first computer I found myself wanting a monitor, a disk drive, anything to make it fully functional. I didn't just want a mantle piece; I wanted to actually use my vintage machine.

After spending some time with my Apple I remembered other computers I had seen in magazines but couldn't afford at the time. I little searching and I was able to start a nice little collection of home computers. I was addicted to vintage computing and collecting everything I could get my hands on.

I was really proud of the fact that every machine I owned, not only worked but I had software to run on them. I was fast becoming an expert in the area of software conversion from the modern computer realm back to actual vintage media like 5 ¼" disk.

Feeling proud that I had a nice collection going I stumbled upon the www.applefritter.com website. I felt pretty versed with apple II hardware and I noticed a small forum there and poked in to read some of the posts. I didn't see anything out of the ordinary but I noticed a subject right above the Apple II section labeled Apple I. Well, I had never seen an Apple I and had only herd the stories like most people about how Woz and Jobs started Apple Computers from their garage. I had to read more.

I quickly learned that only about 50 Apple 1's remained in existence from about 200 boards and they were highly collectable and way out of my price range. I read a post where people talked about making a replica but nobody was really stepping up to the plate. I have an education in engineering so I decided to take a look into the Apple 1 hardware.

In March 2002 I wrote to the Applefritter forum requesting some information on the Apple 1 and I informed the group of my intentions. I was well received and many former and current owners of Apple 1 computers gave me information and documentation to get started.

With manuals and emails from owners in hand, I looked over every inch of the Apple 1. I quickly decided not to use all the original components from the Apple 1 due to their lack of ability. Although there are chip suppliers that carry surplus components, their prices and minimum orders are just out of the question. This left me with a decision that many people questioned. I could have made the replica 1 look like the Apple 1 but in the spirit of Woz, I choose to go with more modern components and reduce the chip count as low as possible.

I quickly found that I could interface 32K of RAM with no address decoding. Simply using Address line A15 as the select line, I could activate the RAM chip with A15 off. So, if the replica 1 was looking anywhere in the address range 0000-7FFF A15 would be low and RAM chip select activated. If the replica 1 was looking at address range 8000-FFFF A15 would now be high and the RAM chip select would be deactivated. I added some logic so that the access time of the RAM write cycle was only during PHI2 of the 6502's clock. For this I used a 74LS00 NAND gate IC. I needed to invert the R/W signal and AND in PHI2. This required 2 of the 4 NAND gates in the IC.

My next step was to add some address decoding so the replica 1 accessed the I/O chip 6821 PIA and ROM at the same addresses the Apple 1 used. I used a 74LS138 to decode addressing for both the 6821 and an 8K EEPROM. I also made sure that the space for BASIC (E000-EFFF) was EEPROM so BASIC could be stored in ROM.

I now had a core CPU section replicated from the Apple 1. I still had no video section and I needed to build a prototype. After some delays I finally built a prototype on wire-wrap and tested it. It didn't work. I needed to troubleshoot the board but my free time was very hard to come by and the replica sat on the shelf for a few months.

In November of 2002 I received an email from Leander Kahney of Wired Magazine. He had been reading on the replica project and wanted to interview me. My first reaction was, "Why me?" I had no idea the cult following that Apple had. I really was fond of Apple since the beginning but Apple has such loyal owners and they love everything that involves the history. So, here I am on the phone with Leander as he is asking me questions about why I am doing the project, and when will it be completed and what will I be selling these for? What? I didn't have the heart to tell him that the project is on the self, that I have no idea what to sell these for even if I had a working version. Little did Leander know that because of him, the replica 1 was pulled off the shelf and completed. I started from scratch, and created a breadboard version of the replica 1. This time it worked... Sort of... Well, the I/O chip, the 6821 was sending binary code for a "\" which means that the replica was seeing a reset and the Woz monitor was trying to output a "\" to a TV monitor. Now I needed a video solution.

Right after the Wired article was released, I received an email from Greg Glawitsch. He stated his interest in the project and said that he had a "down and dirty" solution to the video problem. His answer was to use a microcontroller to output TV text like the video terminal section of the manual. He offered to help by writing the firmware to do this and it would only require a couple external IC's to help the microcontroller. Wow, 3 chips to do what took about 24 on the Apple 1. This was perfect. And with some work with Greg, we had a working video section.

In August 2003 the first of the replica prototypes were being made. I didn't have a name for the board yet and simply called it A1 replica. I also just put my initials on the board with the year 2003. Shortly after my first prototypes were built, I created Briel Computers. At this stage my old high school friend Chris contacted Woz through his webmaster asking for permission to use the Apple 1 code. His approval of the project prompted another Wired article.

In October 2003, the first replica 1's was being built and thus completing a 20 year old dream to design and sell my very own computer. Thanks to Woz for being such a strong influence and great guy to give the ok for this project.

Vince Briel

Setup and Users Manual

Index

Chapter 1:	A brief history of the Apple 1
Chapter 2:	Introducing the replica 1
Chapter 3:	Unpacking and setting up
Chapter 4:	Installing USB drivers
Chapter 5:	Assembling kit version
Chapter 6:	Programming the replica 1
Chapter 7:	Using serial and USB Interface
Chapter 8:	Using the Krusader Assembler
Chapter 9:	Troubleshooting your replica 1
Appendix A:	ASCII Keyboard pin out
Appendix B:	Onboard jumper settings
Appendix C:	Replica 1 Memory Map
Appendix D:	Monitor Listing

Chapter 1: A brief History of the Apple 1

Steve Wozniak had designed and built many different little projects before the Apple 1 computer. The Apple 1 was not even his first computer design. He had designed the “cream soda computer” long before the Apple 1. He had even designed a TV terminal that could be hooked up to a mainframe computer using a keyboard and your TV as the display. Of course he also built the infamous “blue box” so people could phone phreak and make free long distance calls. It wasn’t until Woz decided to combine two projects, and make one complete computer never seen on the market before. He designed a single board computer and combined his TV terminal to create what is today known as the Apple 1 Computer. He first presented the project to the Homebrew Computer Club and it was well received. At that time, the only major computer out there was the Altair 8800 and that only came with LED’s and switches. The Apple 1 had a keyboard and a monitor. When he showed this to his friend Steve Jobs, Jobs was excited and immediately wanted to make more to sell.

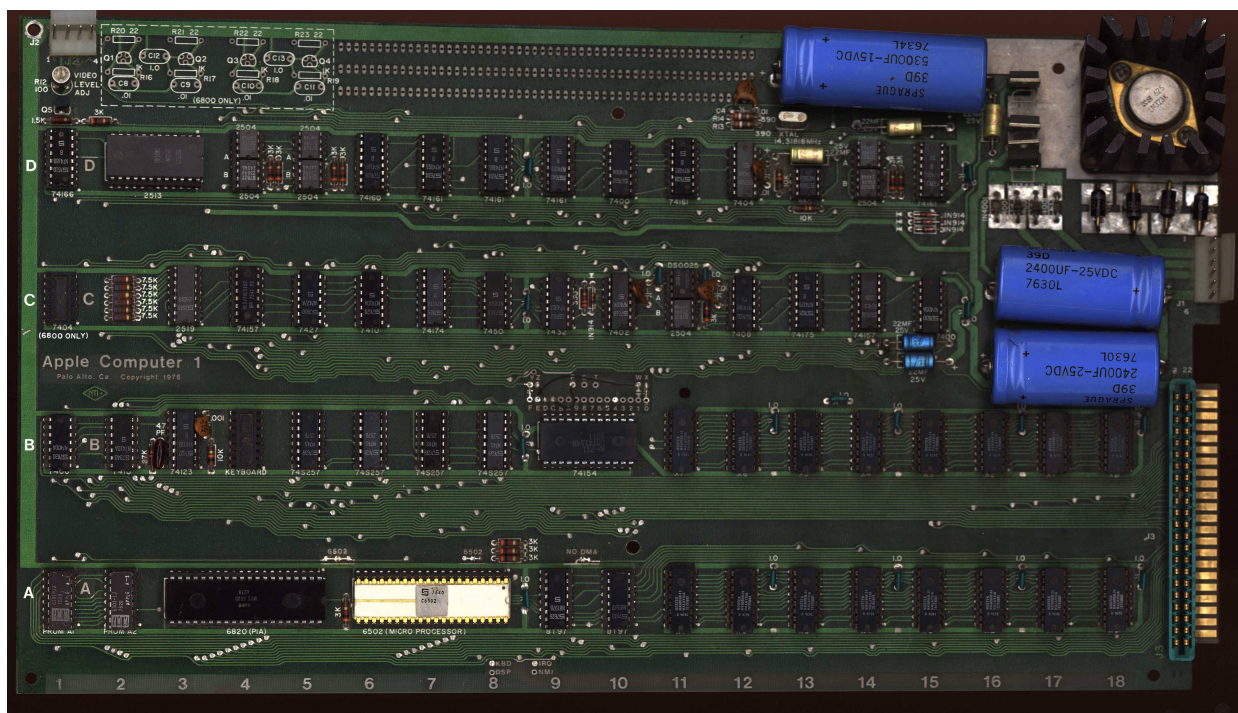


Figure 1: Actual Apple 1 Computer

In April 1976, Steve Jobs and Steve Wozniak decided to start Apple Computer Company. It was based on a computer design that Wozniak had completed making a computer that could display text on a TV and use a keyboard without any additional boards. One board could do all of this. While it wasn’t the first home computer, it was the first to give you all of these features on one board without any addition hardware. No entering code by flipping switches and looking at LED’s. You could actually sit down at a table and type code into the computer using a standard keyboard. Steve Jobs saw something great in this project and was eager to start selling these boards.

Woz distributed schematics and source code listings for the Apple 1 computer at the Homebrew Computer Club meetings in Northern California bay area. Steve Jobs knew that there was enough interest in the boards at the Homebrew club to sell boards to the members. 200 boards were printed and most of them were assembled but not all of them. They assembled boards at night in Job’s parents’ garage and sold some to the Homebrew Club. It wasn’t until a local computer store purchased \$50,000 worth of boards that anything significant had happened. After that, things changed.

They knew they had a great product and that the average person could have a computer on their kitchen table. Woz quickly worked on a BAIC for the Apple 1 but it still lacked a means of storage. Shortly after the Apple 1's release, Woz designed a cassette interface simply named the ACI (Apple Cassette Interface). This interface allowed the user to store programs onto an audio tape using simple 1 and 2KHz signals to store the data.

While the new interface was being designed and BASIC was being written, the end of the Apple 1 was already near. Woz was already working on the next version of his design. While many people thought the Apple II was the same as the Apple 1 with just more features, the Apple II was a completely new design from scratch. This new design could do color and using processor accessible RAM; you could do graphics and more. While this was all being designed, the Apple 1 was still being made and sold.

BAIC was finally released and that gave the Apple 1 a high level language for people to program in. This BAIC was not only written for the Apple 1, but it was also written for the Apple II at the same time. As a matter of fact, if you look at a disassembly of the Apple 1 BAISC you will see some commands built in that are impossible to perform, such as COLOR. There was no color graphics on the Apple 1, only text.

As soon as the Apple II was release, Jobs wanted to get rid of all the Apple 1 computers. He offered deals, like half off an Apple II with the trade in of your old Apple 1. He so wanted to get rid of the Apple 1, that rumor has it, he even gave some Apple II's for even exchanges. Some he handed to engineers and programmers at Apple to use, others he chopped up with a band saw. So, while there were 200 boards made from the board house, not all were assembled and many were destroyed on the exchange program.

Today, approximately 50 Apple 1's are known to exist making it one of the most sought after vintage computers. Some auctions have fetched upwards of \$50,000 for an Apple 1 but the average selling price is around \$16,000-\$25,000.



Figure 2: Steve Jobs and Steve Wozniak shown with an Apple 1 computer in 1976

Chapter 2: Introducing the replica I computer

The replica 1 computer was designed to give those people a chance to see what an Apple I computer looked and acted like. Most people have never seen an Apple I running and the replica 1 helps us see what Woz had created. The goal of the replica 1 project is to simulate the functional aspects of the Apple I so that all hardware and software designed for the Apple I can be used.

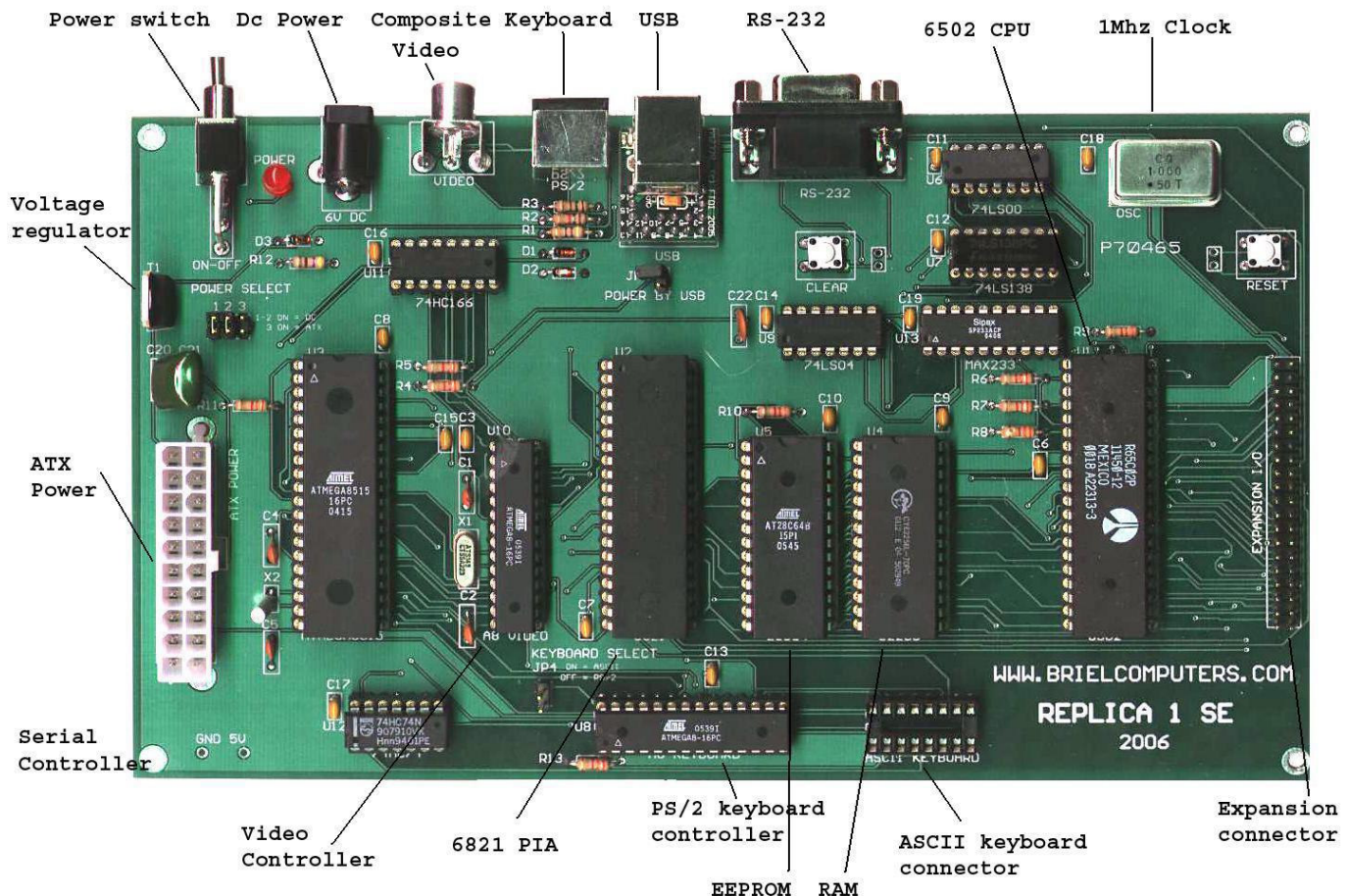
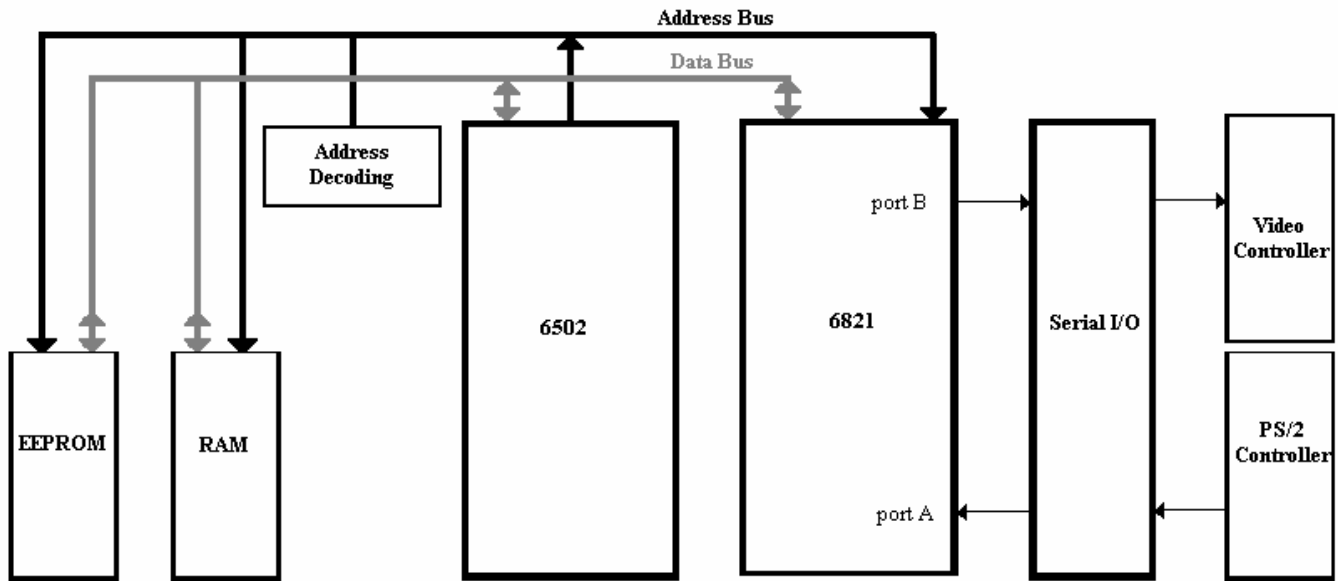


Figure 3: Key component features on the replica 1 SE

Not just a simple hobby board. The replica 1 contains powerful features such as USB (optional) and the ability to have the replica 1 powered by the USB port. Other features include an ATX power supply so that all the voltages on the Apple I are available, a PS/2 keyboard port for those who don't have ASCII keyboards, and a simple On/Off switch for applying power to the replica 1.

The expansion header was designed for the hobby engineer in mind. The expansion connector gives the user the option of using the optional slot expander board for Apple I compatible boards, or running the connector via a ribbon cable to a breadboard for designing add-on hardware.

As well as being able to run all the original Apple I software and hardware, the replica 1 has the capability to use modern hardware like a ps/2 keyboard and use a standard PC ATX power supply or wall brick power supply to power the replica 1. The added USB/serial port gives the user the ability to download programs right from your PC or Mac into the replica 1 making software sharing possible.



replica 1 Block Diagram

Figure 4: Logic diagram of replica 1 SE computer

Keeping a simple design structure was important with the replica 1. Chip reduction is the way Woz designed the Apple 1 and the replica 1 follows in its path. While it's true that the replica 1 board contains more IC's than the original replica 1, the added features give the user flexibility that wasn't available before.

While the replica 1 has been in production for 4 years, it is always a work in progress. Sometimes bugs and fixes come up and changes to the board are always discussed on the replica 1 forums. For the latest in design changes and modifications, see the replica 1 forum at:

<http://www.brielcomputers.com/phpBB2/>

The replica 1 SE is currently the most advanced replica Apple 1 produced. It encompasses features as requested by current or future owners of replica 1, as well as design improvements, bug fixes and other enhancements. All revisions remain backward compatible with the original replica 1.

Chapter 3: Unpacking and Setting up system.

Before you begin setting up your replica 1 computer, make sure you have plenty of table space. Do not power up the unit while touching any part of the circuit board and contact Briel computers if you are unsure about any part of the setup process.

You will need the following items to get your replica 1 up and running:

1. The replica 1 system board.
2. ATX power supply or 6-9V DC wall brick supply positive center.
3. Keyboard, a PC PS/2 or an Apple II/II plus (not //e or //gs).
4. A TV with composite video in, or a composite computer monitor.
5. A video cable to attach to the TV from the replica 1.

Place or mount the replica 1 board so that it is secured. Make sure no metal objects can touch the circuit board or damage could result. Next you need to choose what type of power supply you are going to use. There are three ways you can power your replica 1 SE. ATX power supplies are the most commonly used power supplies in today's computers. You need to use an ATX power supply if you are planning on using the Apple II ASCII keyboard or the Cassette Interface because -12V power is required for the keyboard.



Figure 5: Jumper Select 1-3 shown ready for DC wall brick (not included)

If you only intend on using the ps/2 style keyboard you only need a 7-9V wall brick style power supply. While the replica 1 shows 6V on the board, it is best to use something around 7-9V 1000mA (1.0A) for best performance. To set the jumpers between the two different types, locate the power jumpers 1-3 on the board near the power switch.



Figure 6: DC wall brick connector (Use 7-9V positive center connection power supply)

To use ATX remove any jumpers from 1 and 2 and make sure jumper 3 is on. For DC wall brick power supply (not included), place jumper blocks on jumpers 1 and 2 and remove the jumper from 3.

CAUTION: SETTING THE JUMPERS INCORRECTLY CAN DAMAGE YOUR REPLICA 1. MAKE SURE YOU HAVE THE CORRECT SETTINGS BEFORE YOU POWER UP YOUR REPLICA 1.

The third option for supplying power is to use the power from USB option. If your replica 1 has the optional USB interface, your replica 1 can be powered by the USB port on your PC or Mac.



Figure 7: Jumper Select Power by USB located near optional USB adapter

To use this option, make sure that no other type of power supply (ATX or DC wall brick) is attached to the replica 1. Simply attach the jumper block to enable this option. Next, attach the USB cable to the PC or Mac and the replica 1 will power up.

CAUTION: USING THE POWER BY USB FEATURE WHILE USING THE ATX OR DC WALL BRICK SUPPLY COULD DAMAGE YOUR PC OR MAC OR YOUR REPLICA 1. USE CAUTION WHEN USING THIS OPTION.

Once you have the power source connected, you can attach your monitor or TV to the replica 1 with a standard composite video cable. Locate the video connector as shown and plug your cable (not supplied) into the TV then into the replica 1 (figure 8).



Figure 8: Video connector

Next you need to attach a keyboard. You have two choices for keyboards; ps/2 pc style keyboard or an ASCII keyboard. To use the ps/2 style keyboard locate the ps/2 port on the back edge of the replica 1 and plug your keyboard into that port (see figure 9).



Figure 9: ps/2 keyboard port

If you are using an Apple 2 or 2+ ASCII keyboard, locate the ASCII keyboard port near the replica 1 logo on the board and plug your 16pin dip cable into the board. Note that pin 1 is next to the 'A' on the ASCII KEYBOARD label. Be sure to put a jumper on JP4 to disable the ps/2 keyboard encoder, if you are using an ASCII keyboard. Failure to put a jumper on JP4 will result in the ASCII keyboard not responding.



Figure 10: ASCII keyboard port

Now you are ready to test your replica 1 SE. Make sure your TV or monitor is turned on and flip the switch on the replica 1 up. The red LED next to the switch should light up and a solid block cursor should appear in the upper left corner. If you do not see the cursor at the top of the screen, or the LED is not lit, turn the replica 1 off and recheck your work.

Press RESET on the board and your replica will respond with a backslash “\” and the cursor will drop to the next line. The cursor is a blinking ‘@’ symbol just like the original Apple 1 computer. You are now in the Woz monitor and ready to use your replica 1 SE.

Chapter 4: Installing the USB drivers

If your replica 1 is equipped with the optional USB interface, you need to install drivers on your PC or Mac to use your USB adapter. The drivers are included on your CDROM. The drivers will create a virtual COM port on your PC or Mac so that you can transfer programs to/from the replica 1. Please refer to the CDROM included with your replica 1 SE for the driver install manual for your operating system. If you don't see your operating system listed, check the FTDI website for drivers here:

<http://www.ftdichip.com/Drivers/VCP.htm>

And here for the manuals on installing your drivers:

<http://www.ftdichip.com/Documents/InstallGuides.htm>

Please refer to your computer systems hardware (Mac, Windows XP, etc) for detailed instructions on installing the drivers. Keep in mind that Briel Computers is not the designer of the FTDI USB to serial converter and any technical questions should be directed to FTDI directly.

You will need a standard USB device cable to attach the replica 1 SE to your computer (not included). Once you have the replica 1 SE attached to your computer you can install the drivers. Once the drivers are installed, the drivers will then create a virtual COM port on your machine. Once that is finished you are finished with the driver installation.

Once the drivers have finished, you can check device manager in Windows© and it should look something like figure 11 with the replica 1 attached.

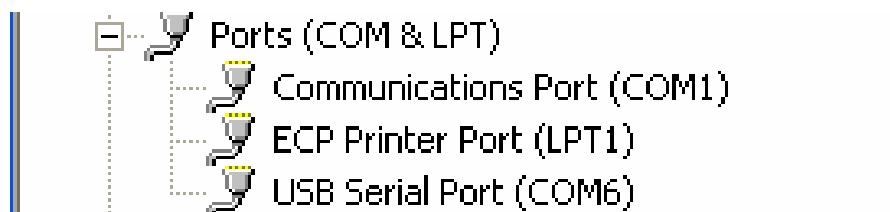


Figure 11: Device Manager showing new USB Serial Port (COM6)

Please contact Briel Computers if you have any questions or problems installing the drivers at vbriel@yahoo.com

Chapter 5: Assembling the replica 1 SE kit

If you have experience with soldering circuit boards, then not only will you enjoy the replica 1 kit, but you will also have the pride in assembling it yourself while saving money. If you have never built a circuit board kit, please start with something a little simpler before attempting to build the replica 1. This chapter is only designed as a brief guide to help you assemble your replica 1.

Kit contents:

All kits have been carefully assembled to insure that no parts are missing; however, you should check the inventory to make sure you have everything. If you find anything missing, please contact Briel Computers.

IC's:

QTY	ID	# of Pins
1	6502	40
1	6821	40
1	62256	28
1	28C64	28
1	ATMEGA8515	40
2	AMEGA8*	28
1	74LS00	14
1	74LS04	14
1	74LS74 was 74HC74	14
1	74LS138	16
1	74HC166	16
1	7805	3
1	MAX233	20

*Atmega8 with white dot is for video, plain is for keyboard

Sockets:

QTY	# of Pins
3	40 pin
2	28 pin (.6 wide)
2	28 pin (.3 wide)
1	20 pin
2	16 pin
1	16 pin duel swipe
3	14 pin

Crystals and Oscillators:

QTY	Type	Size
1	Half Can	14.31818MHz (X1)
1	Cylinder	7.3728MHz (x2)
1	Full Can	1MHz

Connectors:

QTY	Type
1	ATX Male connector
1	RCA style video connector
1	PS/2 keyboard connector
1	2.1mm DC power connector
1	DB-9 Female serial connector
1	40 pin .100" expansion header
1	6 pin jumper header J1-J3
2	2 pin jumper header for J4 & J5
3	Jumper shunts

Switches:

QTY	Type
2	PCB board-mount mini momentary push button
1	PCB board-mount toggle switch

Resistors:

QTY	Size	ID
1	470 ohms	R1
1	1.5K ohms	R2
1	100 ohms	R3
9	3.3K ohms	R4-R11, & R13
1	220 ohms	R12

Diodes:

QTY	Type	ID
3	1N4148	D1-D3

Capacitors:

QTY	Size	ID
4	22pF	C1, C2, C4, C5
15	.1uF	C3, C6-C19
2	.01uF	C21 & C22
1	.22uF	C20

DISPLAYS:

QTY	Type
1	Jumbo Red LED

Circuit Board:

QTY	Type
1	8.5"X5" replica 1 SE silk screened PCB

Tools required for assembly:

Soldering iron, solder, wire cutters, needle nose pliers, and a multi-meter

Before you begin:

Make sure you have proper working space and that you heat your iron before beginning the assembly. This kit is not for beginners and Briel Computers assumes that you have some prior experience with soldering circuit boards. If you have never soldered circuit boards before or don't feel you are ready, you should build other kits of a smaller, less complex nature before building your replica 1.

Reading and identifying resistors:

In previous revisions of the replica 1 kits, all parts were labeled in an effort to make it easier on the kit builder. Now, the manual is being expanded to help newcomers to kits learn to identify the parts and read their values. Resistors have 4 colored bands on them that are used to identify the value. The first 3 are the value and the 4th colored band is to identify the tolerance of the resistor. The 4th band is usually silver for 10% or gold for 5% tolerance. Here is how to read the other three to get the value:

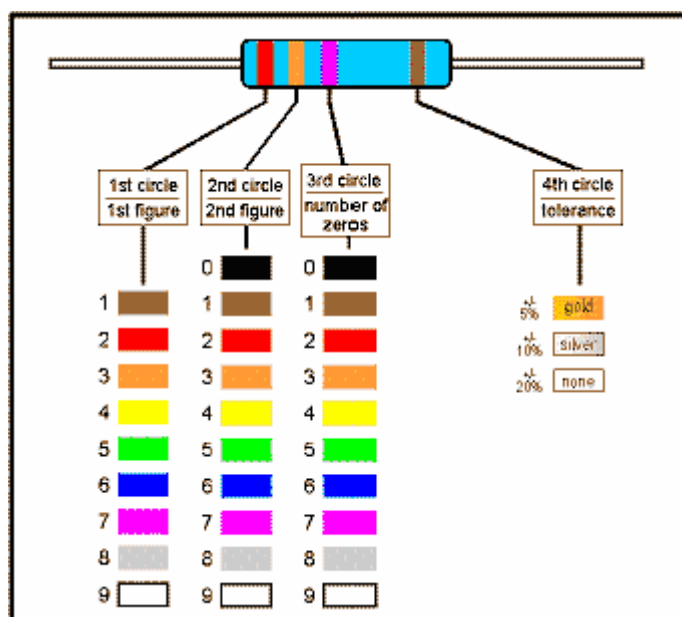


Figure 12: Resistor color code chart

Let's assume the resistor you are trying to identify has the colors orange, and then another orange, then a red band followed with a gold band. So, to read the resistor, find the gold or silver band, that identifies the 4th band. Hold the resistor so the 4th band is on the right side and read the first color on the left.

The first color is orange which has a value of 3 according to the chart above. The second color is also orange which also has a value of 3. The 3rd number indicates how many zeros follow the first two numbers. So, red is 2 zeros. This gives your resistor a value of 3, 3 and 2 0's (00) or 3300=3.3K. This may take some practice but look through all the resistors and identify them without a meter if you need practice.

Reading and identifying capacitors:

It takes less time to read the values of the capacitors than to read the values of the resistors. The key is to understand and read the numbers correctly.

Larger capacitors may have the values clearly printed on them. For example:

10uF = 10 Micro Farads

Smaller capacitors however, may only have 2 or 3 numbers on them. For example, the 22uF capacitors say 22J. .1uF capacitors have the value 104 on them. This is similar to the resistor codes where the first 2 numbers are actual numbers and the 3rd number is the multiplier code. So 10 X 4 is 100,000 and the value is in pico Farads (pF). Converting it to a proper numbering format gives us the value .1uF. To make things easier, here is a list of the values on the capacitors used and the numbers printed on them.

Value	Identifying numbers	QTY
.01uF	103	2
.1uF	104	15
22pF	22J	4
.22uF	224M	1

Diodes:

The diodes used on the replica 1 are to keep voltages from flowing in the wrong direction. They must be installed in the correct orientation. If you look at the PCB for one of the diode locations D1, D2 or D3 you will see a small band inside the diode silkscreen. This indicates which side the black band on the diode goes to. If you don't install these properly, the circuit will not work at all. Check your direction band before installing.

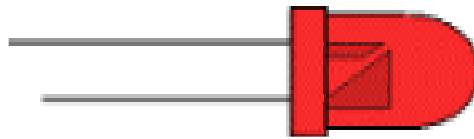
Power LED:

Figure 13: LED pin internal diagram

Looking at the LED there are two connectors, the Anode is longer and has the smaller connector inside. The cathode is the negative side. It has a shorter lead and a larger part inside the LED. The cathode is the negative post. Connecting is simple on the replica 1. The LED silkscreen on the replica 1 has the positive + marked. Simply put the longer anode connector into the positive side when installing.

Ready to Begin Assembly:

Now that you have inventoried all parts you are ready to begin assembly of your replica 1.

Step 1: Start with the IC sockets first. I prefer to start with the larger sockets and work my way down to the smaller ones. Identify the notch on one edge of the socket and match it with the notch on the white silkscreen on the board. Place the socket in its position and flip the board over while holding the socket in place, and rest the board on your table with the socket being pushed through by the table. Make sure it is even and level and solder into place. Repeat this process for all of the sockets.

Step 2: Solder in the momentary 'RESET' and 'CLEAR' buttons. You may need a little tape on the top side to hold the buttons in place.

Step 3: Install the 14.31818MHz (X1) and 7.3728MHz (X2) crystals. The pins may be bowed out a little to hold the crystal in place. There is enough room to fold the 7.3728 MHz crystal flat away from the 40 pin socket.

Step 4: Install all resistors, capacitors and diodes. When installing, bend the leads outward to keep component from falling out when flipping the board over to install. Clip the extra lead length off after soldering.

Step 5: Install LED as described earlier. Remove extra lead.

Step 6: Install 1MHz oscillator, 1 corner is pointed, that is pin 1 (matches PCB silkscreen outline)

Step 7: Install the 40 pin expansion connector (notice there is no incorrect way to install but pin 1 is closer to the reset button.

Step 8: Install power switch, video connector, DB-9 serial connector and ps/2 keyboard connector.

Step 9: Install jumper post. JP1, JP2 and JP3 are one piece near the power switch. JP4 is near the 28 pin socket for the Atmega8 keyboard controller. JP5 is near the USB controller.

Step 10: Install the voltage regulator 7805 near the power switch labeled T1. Make sure metal side faces board edge.* Install the ATX connector.

*Installing the 7805 in the wrong direction may damage your replica 1 and the circuit board will not function.

Step 11: Install the optional USB adapter. You may wish to debug and test your board before installing the USB adapter. This will help reduce risk of damage if you made an error in soldering. There are two ways to install the USB adapter; you may choose to use the mounting posts, and then plug the USB adapter into the posts. Or, you may choose to skip the posts and install directly onto the replica 1 for a cleaner look. Either way works just as good.

Inspection of your work:

Now that you have completed your soldering, take a few minutes and look over your work. Check for solder spots that might not be good connections to the posts. Look for bridges where solder may have jumped over where it wasn't supposed to go.

Set up power supply jumpers:

When everything looks good, set the jumpers for the type of power supply you are using. If you are using an ATX, set jumper 3 and leave off shunts from JP1 and 2. If you are using the DC wall power supply (not included) set jumpers JP1 and JP2 and leave JP3 off. If you are going to use the power supplied from the USB interface, make sure you don't have a DC wall brick power supply or ATX attached, and put the shunt on JP5.

CAUTION: USE THE POWER BY USB OPTION WITH EXTREME CAUTION. ANY ERROR IN THE POWER SUPPLY JUMPER SETTINGS WILL DAMAGE YOUR REPLICA 1 AND POSSIBLY YOUR COMPUTER.

Set keyboard jumper:

If you have a 7 bit parallel ASCII keyboard you can use it with the replica 1. See Appendix A for a diagram of the pin out of the ASCII keyboard port. If you are using the ASCII keyboard option, place a shunt on JP4 so that the ps/2 controller doesn't interfere with the ASCII keyboard operation.

Testing your replica 1:

Before installing your IC's, power up your replica 1. With your multi-meter, check the voltages 5V and GND. make sure your LED is lit properly. Test pin 14 and pin 7 on the 74LS00, 74LS04, and 74LS74 to make sure you are getting 5V. Check pin 8 and 16 on 74LS138 and 74HC166 for 5V. Once you are satisfied that the power is being distributed correctly, power off the replica. Install the Atmega8 video IC (labeled on the bottom with a white spot), the 74HC74 and the 74HC166. Attach an RCA cable to the replica 1 video port and attach it to your monitor or TV. Power up your replica 1 and see if you get a cursor. It may move across the screen and that is normal since it is getting a false trigger. If the video section is working properly, turn off the replica 1 and install the rest of the chips. Power up the replica 1 again, this time press 'RESET' button after a couple of seconds and see if the replica 1 resets.

If the circuit is working properly, you will get a “\” backslash followed by the cursor ‘@’ moving to the next line. At this point you can power off the replica 1 once more and attach your keyboard. Power up the replica 1, reset the circuit and try out the keyboard. Congratulations, your replica 1 is now assembled and ready to use.

Caution:

Before plugging in the power supply, keyboard and video connector, thoroughly check over your work for errors.

Any parts that get accidentally damaged during the building process can be replaced at the owner’s expense. Contact Briel Computers for any replacement parts needed.

The goal of Briel Computers is that ALL replica 1 kits get completed successfully. If you are having a problem, please feel free to ask for help.

Use the rest of this page to document any notes needed for assembly

Chapter 5: Programming the replica 1

The replica 1 can be programmed in one of 3 built in ways. You can enter values into memory directly using the Woz monitor. Another way is to use BASIC that is built into ROM. BASIC was the most commonly used language among beginner all the way to experienced programmers in the 70's and 80's. The third built in way to program your replica 1 SE is to use the new Krusader© assembler written by Ken Wessen.

Using the Woz monitor:

The Woz monitor is a tool that gives you direct access to memory to program code that the 6502 will interpret. To use the Woz monitor program, simply power up your replica 1. Just like the original Apple 1© you had to manually reset the circuit to get the computer started. Simply press the reset button on the circuit board and you will see a backslash '\ ' and the cursor will drop below the backslash. You are now in the Woz monitor.

There are three basic functions to the Woz monitor. First, you can examine a range of memory or just one location. To examine one location of memory, just type in the address and press [return]. To view memory location 300 it would look like this:

```
300[RETURN]
```

```
0300: E1
```

The cursor will then move below the line that displayed the value of the memory location. To display a block of memory, type in the starting location followed by a period '.'; follow that with the ending location. To view 300-32F locations, type:

```
300.32F[RETURN]
```

The replica will display the memory contents up to 8 locations per line:

```
0300: E1 20 94 17 08 26 60 10
0308: E4 98 02 86 88 44 A5 50
0310: 3C 2D 20 9D 7E 18 60 8C
0318: 0F 00 18 24 90 24 06 50
0320: 42 07 06 A0 89 81 00 C6
0328: 80 60 41 12 E0 1B A3 04
```

Again, the cursor will be below the listing.

To program a location in memory, simply type in the address followed by the value to write to it. For example, to program location 300 with the value FF, simply type:

```
300:FF [RETURN]
```

The Woz monitor will respond by displaying the old value of the memory location:

```
0300:E1
```

The monitor will once again drop below the displayed line and be ready for more input. To program more than one consecutive location simply put a space after the value, then type in the next value of the next memory location. For example, to program 300 with FF, 301 with EE and 302 with DD type the following line:

300: FF EE DD [RETURN]

The monitor will respond with the prior contents of the first location you programmed (300):

0300:E1

If you examine the other two locations in memory, you will see the new value is now stored. Type:

301.302[RETURN]

The monitor then displays:

0301: EE DD

Notice it did not display 8 locations of memory across because you only asked for two. Once you have entered your program into memory, you can start your program by entering the memory location that it starts at followed by **R** for run and [return]. Let's load and run a simple program through the Woz monitor. Type in the following code:

300: A9 0 AA 20 EF FF E8 8A 4C 2 3 [RETURN]

Notice the '3' and the '2' are single digits, this is correct. You can type in '02' or '00' but it is not necessary. After you have typed in the line, type:

300.30A

This will now print out the contents of the memory locations you just typed in. Verify the contents match the code, then type:

300R [RETURN]

The program should then print out a continuous stream of characters. Every once in a while the screen will clear; this is a normal function of the replica 1 having the ability to clear the screen through software.

Programming in BASIC:

To start the replica 1 and enter BASIC, turn on the replica and press reset. Next type in:

E000R [RETURN]

The memory contents of E000 are displayed and BASIC begins. The prompt for BASIC is '>'. This is how you know you are in BASIC. Here is a simple 'HELLO' program you can type in:

```
10 PRINT "HELLO WORLD!"  
20 END
```

RUN

When you type in **RUN**, the program will print out "HELLO WORLD!" onto the display, and then it will return to the prompt. You can view your program by typing in "LIST". To get started programming in BASIC, read the BASIC manual. There are many example programs on the internet to help you get started.

Krusader:

The replica 1 has 8K of EEPROM (ROM) space onboard. 4K is used for BASIC; 256 bytes are used for the Woz monitor. This leaves just under 4K available. Now, thanks to Ken Wessen, that space is now occupied with a powerful assembler called Krusader. To start Krusader, type in '**F000R**' [return]. To program the assembler, refer to the Krusader manual for sample programs and commands.

Chapter 5: Using the serial and USB interface

Setting up Hyper Terminal:

Start a new session of Hyper Terminal and select a name for your session.



Figure 14: Setting up a new connection

Click on OK; next, select which COM port you are connected to and select OK. Next, set up your COM port as shown: 2400, N,8,1

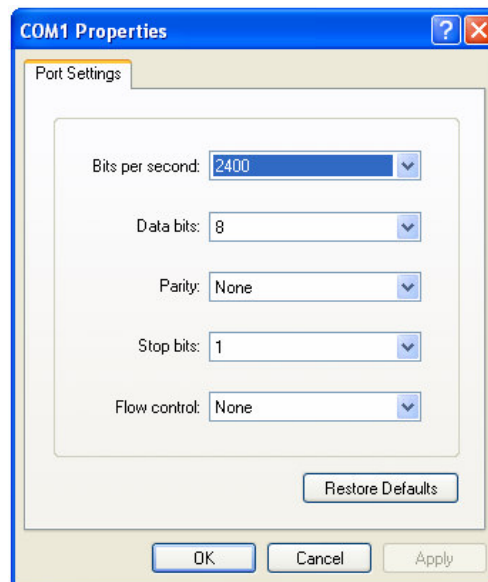



Figure 15: Proper serial port settings

Select OK and continue on to starting Hyper Terminal. You are not ready yet, so click on the disconnect Icon as shown:  Next, click on “File” in the menu bar and select “Properties” from the drop-down menu list. You should then get a screen that looks like this:

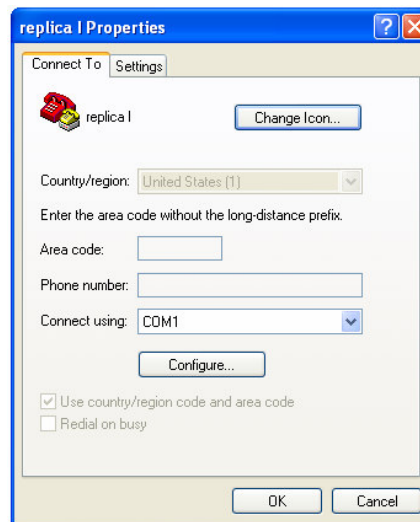


Figure 16: Configure window

Click the “Settings” tab at the top, and then select the “ASCII setup” button.

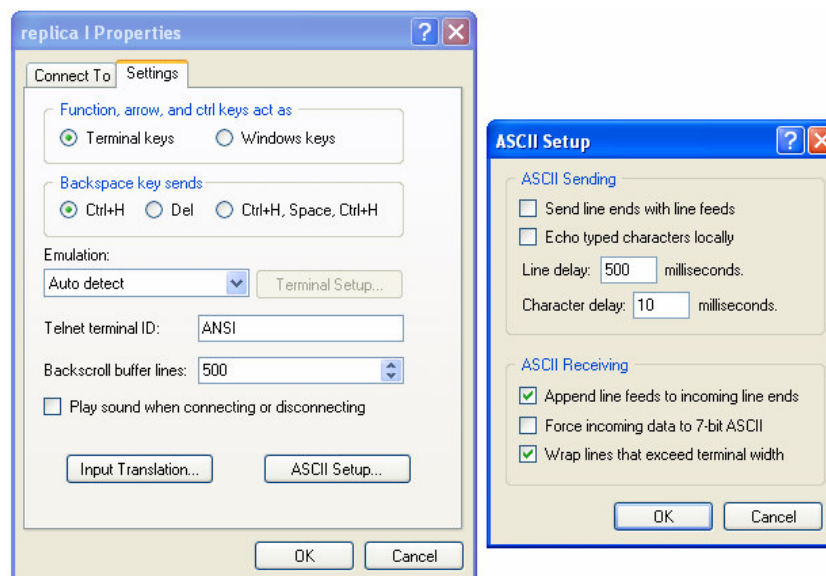



Figure 17: ASCII settings for COM port

Set up your settings exactly as shown in the Figure 13. Select “OK” to close the ASCII Setup screen. Select “OK” again to close the properties window and return to Hyper Terminal. Now click on the “Connect” button  and you are connected. Power up your replica and press the reset button. Try typing on both the replica keyboard and your pc keyboard to verify that both function. If you have any problems repeat these steps.

Transferring files with Hyper Terminal

To send a file to your replica, simply click on the “Transfer” tab and select “Send Text File” from the drop down list.

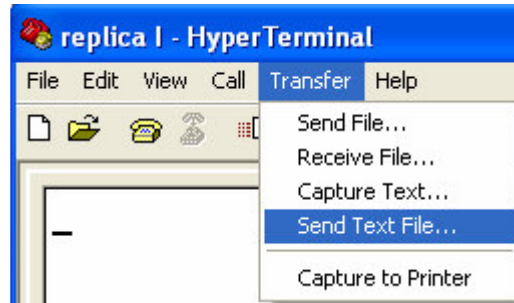


Figure 18: Send text file

Find the file on your PC and select “OK”. At this point Hyper Terminal will send the file your replica and your replica will treat the file as if you were typing it in. Once it is finished typing in, just run your program as normal. Remember, if you are sending a BASIC file to enter BASIC by typing “E000R” followed by “Enter” before sending the file.

To receive a file is just as easy. Once again, click on the “Transfer” tab, but this time select “Capture Text...” from the drop down list. This will bring up a file name to save option and select the name for your file. Select “OK” to begin the “Capture” mode. If you’re in BASIC you can type LIST and Hyper Terminal will capture your BASIC listing. If you’re in the Woz monitor, simply type the address range to capture, followed by “Enter.”

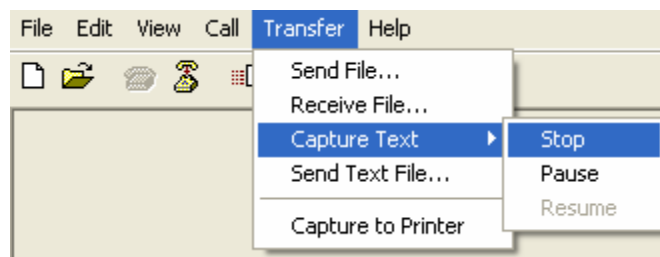


Figure 19: Stop text capture

Once you are done capturing your file, click on the “Transfer” tab, Capture Text option and “Stop.” That is it. You can now go in and edit your file with any text editor.

Chapter 6: Using the Krusader Assembler

Krusader is a powerful assembler written by Ken Wessen for the replica 1. This assembler makes programming the 6502 processor in the replica 1 a breeze. No hand converting mnemonic instructions to hex then manually entering them in the Woz monitor. Now you can program using mnemonic instructions, assemble and run your program right from the assembler. To enter the assembler, type in:

F000R [return]

The program will begin and the authors name and current revision will appear. Next line will display a '?' with the cursor next to it. Type 'N' to enter a new program. The assembler will now display '000' for the first line. Let's enter the program listed in the Krusader as an example program.

000 is now displayed. Press space bar to skip the first field used for naming code blocks. The assembler tabs over to the next field. Type **LDA #'A' [return]**;

001 is now displayed. Then type **LOOP** then press space bar to tab to the next field. Type **JSR \$FFEF [return]**. Type in the rest as shown:

```
002      CLC
003      ADC #$1
004      CMP #'Z'+1
005      BNE LOOP
006      RTS
```

Once you have entered in the source code, type L to list out the code, and then A to assemble it. To run your code type **R \$300**. The program will run and output the string:

ABCDEFGHIJKLMNOPQRSTUVWXYZ

Once the program finishes with the **RTS** command it will return to the Krusader assembler shell.

To learn more on how to program using Krusader, refer to the Krusader owner's manual or visit the Krusader web site at:

<http://school.anhb.uwa.edu.au/personalpages/kwessen/apple1/Krusader.htm>

Chapter 7: Troubleshooting your replica 1

If you have just built your replica 1 and are now having problems getting it running, here are some tips on what to check:

Dead: Check jumpers to ensure correct setting for the type of power source you are using. Check your power source to make sure it is functioning properly.

Replica 1 won't reset: If you soldered from a kit, check for any points that didn't get soldered. Check to make sure all chips are seated properly. Make sure you soldered the oscillator in the correct orientation.

No response from the keyboard: Check keyboard select jumper J4 for correct setting. If using a ps/2 keyboard, try another keyboard. If built from kit, check for missed solder points.

Reset works but cursor stays on same line as backslash: This is an indication that communication between the 6821 and the 6502 is not working correctly, or RAM is not functioning properly. If built from kit, check the 6821, 74LS04 and 62256 for missed soldering of pins. Make sure the chips are seated properly.

These are a few of the most common problems reported by kit builders. The number one source was missed pins. If the pins don't get soldered, the circuit may work, but act very strange. If you absolutely can't find out what is wrong, contact Briel Computers to have your board checked.

The replica 1 forums at: <http://www.brielcomputers.com/phpBB2/> is a good source for information when troubleshooting.

We want every kit that gets sold to operate properly so we will fix your kit for shipping charges plus cost on any parts. Please contact Briel Computers at ybriel@yahoo.com before shipping in your board for repair.

Appendix A: ASCII Keyboard pin out

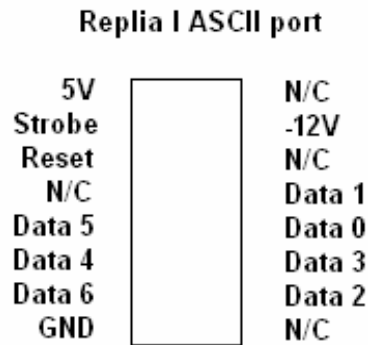


Figure 20: replica I keyboard port from the 16 pin DIP connector. Pin 1 is 5V.

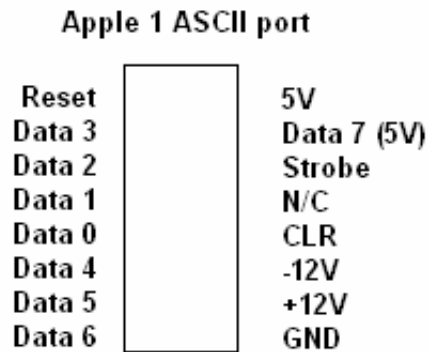


Figure 21: Apple 1 keyboard port from the 16 pin DIP connector; Pin 1 is reset.

Appendix B: Onboard jumper settings

There are a total of 5 jumpers on the replica 1 SE main board. A maximum of 3 can be in use at any time. Three of them (JP1-JP3), are located near the power switch. Jumper shunts are black connectors that fit on top of the jumpers to connect the two pins together for that jumper. To set the power supply up for ATX standard PC power supply, simply place one of the jumper shunts on JP3 and make sure nothing is on JP1 and JP2. If you are using a DC power supply, place jumper shunts on JP1 and JP2 and remove any shunt from JP3.

If you are using USB option and want to power the replica 1 off of your USB connection to your PC or Mac you must first disconnect any DC wall connector or ATX connector from your replica 1. Next place a jumper shunt on JP5. When you plug in your USB cable, the replica 1 will be powered up and the LED will light up.

CAUTION: BE SURE TO TAKE OFF JP5 JUMPER WHEN USING DC OR ATX POWER SUPPLY OPTIONS. FAILURE TO REMOVE JP5 WILL RESULT IN DAMAGE TO YOUR REPLICA 1 AND POSSIBLY YOUR COMPUTER IF ATTACHED TO THE USB PORT.

JP4 is your keyboard source select. Most replica 1 owners use ps/2 keyboards with their replica 1 so no jumper is required. If, however, you are using a 7 bit parallel ASCII keyboard with your replica 1, you need to place a jumper over JP4 to disable the ps/2 controller.

Appendix C: Replica 1 Memory Map

\$00-\$FF Zero Page RAM
\$0100-\$7FFF User RAM for BASIC or Machine Language programs
\$8000-\$8FFF CFFA1 Scratch Memory
\$9000-AFFF CFFA1 Firmware
\$B000-BFFF Unused area
\$C000-CFFF Cassette Interface
\$D000-DFFF PIA I/O Area
\$E000-EFFF BASIC
\$F000-FEFF Krusader \$FF00-FFFF Woz Monitor

Appendix D: Woz Monitor listing

FF00: D8	RESET	CLD	Clear decimal arithmetic mode.
FF01: 58		CLI	
FF02: A0 7F		LDY #\$7F	Mask for DSP data direction register.
FF04: 8C 12 D0		STY DSP	Set it up.
FF07: A9 A7		LDA #\$A7	KBD and DSP control register mask.
FF09: 8D 11 D0		STA KBD CR	Enable interrupts, set CA1, CB1, for
FF0C: 8D 13 D0		STA DSP CR	positive edge sense/output mode.
FF0F: C9 DF	NOTCR	CMP #\$DF	“_”?
FF11: F0 13		BEQ BACKSPACE	Yes.
FF13: C9 9B		CMP #\$9B	ESC?
FF15: F0 03		BEQ ESCAPE	Yes.
FF17: C8		INY	Advance text index.
FF18: 10 0F		BPL NEXTCHAR	Auto ESC if > 127.
FF1A: A9 DC	ESCAPE	LDA #\$DC	“\”.
FF1C: 20 EF FF		JSR ECHO	Output it.
FF1F: A9 8D	GETLINE	LDA #\$8D	CR.
FF21: 20 EF FF		JSR ECHO	Output it.
FF24: A0 01		LDY #\$01	Initialize text index.
FF26: 88	BACKSPACE	DEY	Back up text index.
FF27: 30 F6		BMI GETLINE	Beyond start of line, reinitialize.
FF29: AD 11 D0	NEXTCHAR	LDA KBD CR	Key ready?
FF2C: 10 FB		BPL NEXTCHAR	Loop until ready.
FF2E: AD 10 D0		LDA KBD	Load character. B7 should be ‘1’.
FF31: 99 00 02		STA IN,Y	Add to text buffer.
FF34: 20 EF FF		JSR ECHO	Display character.
FF37: C9 8D		CMP #\$8D	CR?
FF39: D0 D4		BNE NOTCR	No.
FF3B: A0 FF		LDY #\$FF	Reset text index.
FF3D: A9 00		LDA #\$00	For XAM mode.
FF3F: AA		TAX	0->X.
FF40: 0A	SETSTOR	ASL	Leaves \$7B if setting STOR mode.
FF41: 85 2B	SETMODE	STA MODE	\$00=XAM \$7B=STOR \$AE=BLOK XAM
FF43: C8	BLSKIP	INY	Advance text index.
FF44: B9 00 02	NEXT ITEM	LDA IN,Y	Get character.
FF47: C9 8D		CMP #\$8D	CR?
FF49: F0 D4		BEQ GETLINE	Yes, done this line.
FF4B: C9 AE		CMP #\$AE	“.”?
FF4D: 90 F4		BCC BLSKIP	Skip delimiter.
FF4F: F0 F0		BEQ GETLINE	Yes. Set STOR mode.
FF51: C9 BA		CMP #\$BA	“:”?
FF53: F0 EB		BEQ SETSTOR	Yes. Set STOR mode.
FF55: C9 D2		CMP #\$D2	“R”?
FF57: F0 3B		BEQ RUN	Yes. Run user program.
FF59: 86 28		STX L	\$00-> L.
FF5B: 86 29		STX H	and H.
FF5D: 84 2A		STY YSAV	Save Y for comparison.

FF5F: B9 00 02	NEXTHEX	LDA IN,Y	Get character for hex test.
FF62: 49 B0		EOR #\$B0	Map digits to \$0-9.
FF64: C9 0A		CMP #\$0A	Digit?
FF66: 90 06		BCC DIG	Yes.
FF68: 69 88		ADC #\$88	Map letter "A"-“F” to \$FA-FF.
FF6A: C9 FA		CMP #\$FA	Hex letter?
FF6C: 90 11		BCC NOTHEX	No, character not hex.
FF6E: 0A	DIG	ASL	
FF6F: 0A		ASL	Hex digit to MSD of A.
FF70: 0A		ASL	
FF71: 0A		ASL	
FF72: A2 04		LDX #\$04	Shift count.
FF74: 0A	HEXSHIFT	ASL	Hex digit left, MSB to carry.
FF75: 26 28		ROL L	Rotate into LSD.
FF77: 26 29		ROL H	Rotate into MSD's.
FF79: CA		DEX	Done 4 shifts?
FF7A: D0 F8		BNE HEXSHIFT	No, loop.
FF7C: C8		INY	Advance text index.
FF7D: D0 E0		BNE NEXTHEX	Always taken. Check next char for hex.
FF7F: C4 2A	NOTHEX	CPY YSA V	Check if L, H empty (no hex digits).
FF81: F0 97		BEQ ESCAPE	Yes, generate ESC sequence.
FF83: 24 2B		BIT MODE	Test MODE byte.
FF85: 50 10		BVC NOTSTOR	B6=0 STOR 1 for XAM & BLOCK XAM
FF87: A5 28		LDA L	LSD's of hex data.
FF89: 81 26		STA (STL,X)	Store at current 'store index'.
FF8B: E6 26		INC STL	Increment store index.
FF8D: D0 B5		BNE NEXTITEM	Get next item. (no carry).
FF8F: E6 27		INC STH	Add carry to 'store index' high order.
FF91: 4C 44 FF	TONEXTITEM	JMP NEXTITEM	Get next command item.
FF94: 6C 24 00	RUN	JMP (XAML)	Run at current XAM index.
FF97: 30 2B	NOTSTOR	BMI XAMNEXT	B7=0 for XAM, 1 for BLOCK XAM.
FF99: A2 02		LDX #\$02	Byte count.
FF9B: B5 27	SETADR	LDA L-1,X	Copy hex data to
FF9D: 95 25		STA STL-1,X	'store index'.
FF9F: 95 23		STA XAML-1,X	And to 'XAM index'.
FFA1: CA		DEX	Next of 2 bytes.
FFA2: D0 F7		BNE SETADR	Loop unless X=0.
FFA4: D0 14	NXTPRNT	BNE PRDATA	NE means no address to print.
FFA6: A9 8D		LDA #\$8D	CR.
FFA8: 20 EF FF		JSR ECHO	Output it.
FFAB:A5 25		LDA XAMH	'Examine index' high-order byte.
FFAD:20 DC FF		JSR PRBYTE	Output it in hex format.
FFB0: A5 24		LDA XAML	Low-order 'examine index' byte.
FFB2: 20 DC FF		JSR PRBYTE	Output it in hex format.
FFB5: A9 BA		LDA #\$BA	“:”.
FFB7: 20 EF FF		JSR ECHO	Output it.
FFBA:A9 A0	PRDATA	LDA #\$A0	Blank.
FFBC:20 EF FF		JSR ECHO	Output it.
FFBF: A1 24		LDA (XAML,X)	Get data byte at 'examine index'.

FFC1: 20 DC FF		JSR PRBYTE	Output it in hex format.
FFC4: 86 2B	XAMNEXT	STX MODE	0->MODE (XAM mode).
FFC7: A5 24		LDA XAML	
FFC8: C5 28		CMP L	Compare 'examine index' to hex data.
FFCA: A5 25		LDA XAMH	
FFCC: E5 29		SBC H	
FFCE: B0 C1		BCS TONEXTITEM	Not less, so no more data to output.
FFD0: E6 24		INC XAML	
FFD2: D0 02		BNE MOD8CHK	Increment 'examine index'.
FFD4: E6 25		INC XAMH	
FFD6: A5 24	MOD8CHK	LDA XAML	Check low-order 'examine index' byte
FFD8: 29 07		AND #\$07	For MOD 8=0
FFDA: 10 C8		BPL NXTPRNT	Always taken.
FFDC: 48	PRBYTE	PHA	Save A for LSD.
FFDD: 4A		LSR	
FFDE: 4A		LSR	
FFDF: 4A		LSR	MSD to LSD position.
FFE0: 4A		LSR	
FFE1: 20 E5 FF		JSR PRHEX	Output hex digit.
FFE4: 68		PLA	Restore A.
FFE5: 29 0F	PRHEX	AND #\$0F	Mask LSD for hex print.
FFE7: 09 B0		ORA #\$B0	Add "0".
FFE9: C9 BA		CMP #\$BA	Digit?
FFEB: 90 02		BCC ECHO	Yes, output it.
FFED: 69 06		ADC #\$06	Add offset for letter.
FFEF: 2C 12 D0	ECHO	BIT DSP	DA bit (B7) cleared yet?
FFF2: 30 FB		BMI ECHO	No, wait for display.
FFF4: 8D 12 D0		STA DSP	Output character. Sets DA.
FFF7: 60		RTS	Return.
FFF8: 00 00 (unused)			
FFFA: 00 0F (NMI)			
FFFC: 00 FF (RESET)			
FFFE: 00 00 (IRQ)			

HARDWARE NOTES

Page 0 Variables

XAML	24
XAMH	25
STL	26
STH	27
L	28
H	29
YSAV	2A
MODE	2B

Other Variables

IN	200-27F
KBD	D010
KBD CR	D011
DSP	D012
DSP CR	D013

Warranty

(Applies to factory assembled circuit boards only)

Briel Computers hereby warrants each of its products, and all components therein contained, to be free from defects in materials and/or workmanship for a period of thirty (30) days from date of purchase. In the event of the occurrence of malfunction or other indication of failure attributable directly to faulty workmanship and/or material, then, upon return of the product to Briel Computers, at 5392 Cornell Blvd, North Ridgeville, Ohio 44039 (postage prepaid), Briel Computers will, at its option, repair or replace said products or components thereof, to whatever extent Briel Computers shall deem necessary, to restore said product to proper operating condition. All such repairs or replacements shall be rendered by Briel Computers, without charge to the customer. The responsibility for the failure of any Briel Computers product, or component thereof, which, at the discretion of Briel Computers, shall have resulted either directly or indirectly from accident, abuse, or misapplication of the product, shall be assumed by the customer, and the Briel Computers shall assume no liability as a consequence of such events under the terms of this warranty. While every effort, on the part of Briel Computers, is made to provide clear and accurate technical instruction on the use, implementation, and application of its products, Briel Computers shall assume no liability in events which may arise from the application of such technical instruction, nor shall Briel Computers be held liable for the quality, interconnection, or application of peripheral products, which may have been recommended by Briel Computers, but which have not been supplied as part of the product. This warranty contains and embodies the limits of responsibility of Briel Computers, with regard to its products, and no other liability is expressed, implied, or should be assumed by the purchaser, and in no event shall Briel Computers be held liable for the loss of time, effort, or transportation costs, nor for loss of potential profits or other consequential losses which might arise from the purchase, assembly, use, application, or subsequent sale of the products of Briel Computers, nor from any instructions and/or technical information thereto related.