miniFLASH

User Manual

Cloud-9
3749 County Road 30
Delano, MN 55328
Telephone 763.972.3261
www.cloud9tech.com

While reasonable efforts have been made in the preparation of this manual to assure its accuracy, Cloud-9 does not assume liability resulting from any errors in or omissions from this manual, or from the use of the information contained herein.

Table Of Contents

1. Congratulations!

Thank you for purchasing the miniFLASH, another one of the fine products offered by Cloud-9. This product was designed to be used with the Radio Shack/TRS-80 Color Computers 1, 2, and the Tandy Color Computer 3.

The miniFLASH allows you to use FLASH technology to store ROM Pak images or your favorite DOS.

Features

- Uses state-of-the-art components for low power draw and decreased part count.
- 64K of internal FLASH holds four distinct 16K ROM images.
- Easy to use FLASH programming and IDE test software.

System Requirements

In order to use the miniFLASH, you need the following components:

- Radio Shack/TRS-80 Color Computer 1 or 2 with 64K RAM or a 128K Tandy Color Computer 3.
- Multi-Pak required for certain FLASH programming tasks.

2. Getting Started

We know you're excited about using your new miniFLASH! However, before you plug it into your CoCo, please read the following pages carefully for important configuration information.

Examining the Interface

There are a number of DIP switches on the miniFLASH that must be set properly to reflect your particular system

DIP Switches

The DIP switch block has three switches, and can be accessed even with the cover in place. The switches are labeled from right to left as 1, 2 and 3 and perform the following functions:

Switch 1: This is the auto-start switch. If this switch is ON (in the Down position), then the CART interrupt is triggered upon power up. If this switch is OFF (in the Up position) then the CART interrupt is not triggered on power up.

This switch is useful if you have an auto-start ROM, such as a game, in the selected FLASH bank. You should NOT turn the auto-switch on if your FLASH bank contains a DOS, such as HDB-DOS.

Switch 2 and 3: Both of these switches are used to select one of four 16K banks of the internal 64K of FLASH. The following table show the settings:

Switch 2 Position	Switch 3 Position	FLASH Bank
ON (Down)	ON (Down)	0
ON (Down)	OFF (Up)	1
OFF (Up)	ON (Down)	2
OFF (Up)	OFF (Up)	3

Each of the four 16K banks can contain either a DOS (such as Disk BASIC, HDB-DOS or A-DOS), or the contents of a ROM Pak. We will learn later how to burn a ROM image into the FLASH.

Jumper Block

The jumper block allows the setting of the interface's base address, as well the Flash Bank Selection Register (FBSR) behavior on RESET/Power-Up.

There are 5 jumper pairs, for a total of 10 pins. Looking at the pins from right to left, the bottom row of pins is odd numbered (1, 3, 5, 7 and 9) and the top row of pins is even numbered (2, 4, 6, 8 and 10). Each top and bottom pin forms a pair that may or may not be covered with a shunt.

The following table shows the function of each of the five pairs of jumpers:

Jumper Pair	Function if ON	Function if OFF
1-2	FBSR resets on power cycle	FBSR resets on RESET
3-4	Not used	Not used
5-6	Bit 4 of base address is SET	Bit 4 of base address is CLEAR
7-8	Bit 5 of base address is SET	Bit 5 of base address is CLEAR
9-10	Bit 6 of base address is SET	Bit 6 of base address is CLEAR

You will learn shortly how to set these jumpers for your specific system setup.

Configuring the Interface

In order to insure that your miniFLASH matches your Color Computer system, it is important to set the state of the jumpers to reflect the proper base address as well as FBSR reset behavior.

Setting the Base Address

As shipped, the interface's base address is set up at a default of $FF59. We recommend keeping this setting if possible. In order to this, you must verify that no other devices in your system conflict with this address allocation.

If you do have a conflict then a base address must be selected where there are no conflicts. The interface allows a base address selection of $FF49, $FF59, $FF69 and $FF79. Jumper pairs 5-6, 7-8 and 9-10 allow you to select one of these four base addresses.

The following table shows how to set the jumper shunts based on the desired base address. Please note that when a jumper shunt is OFF, the corresponding bit is SET:

Address	J5-J6 (Bit 4)	J7-J8 (Bit 5)	J9-J10 (Bit 6)
$FF49	Shunt OFF	Shunt ON	Shunt ON
$FF59	Shunt OFF	Shunt ON	Shunt OFF
$FF69	Shunt OFF	Shunt OFF	Shunt ON
$FF79	Shunt OFF	Shunt OFF	Shunt OFF

Setting the FLASH Reset Behavior

The Flash Bank Selection Register (FBSR) is located at offset $09 from the base address of the interface. At power-up, this register holds the value of DIP Switches 2-3, which determine which of the four banks of 16K appear in the CoCo's ROM space at $C000. Software can modify the value in this register to automatically select a different bank of FLASH.

If jumper pair 1-2 contains a shunt, then the contents of the FBSR will remain the same even through hardware reset, and only a power cycle will cause the FBSR value to change back to the DIP switch settings.

If jumper pair 1-2 does NOT contain a shunt, then contents of the FBSR are reset to the state of the DIP switches upon pressing the RESET button.

For a CoCo 1 or CoCo 2, we recommend that you keep the shunt ON, since these Color Computers run from ROM. Hence, pressing the RESET button insures that the last value written to the FBSR remains, and that the corresponding FLASH bank stays mapped into memory space.

Connecting to the CoCo

Now, it's time to connect the miniFLASH to your Color Computer. Be sure that power is OFF on your Color Computer and your Multi-Pak, then insert the miniFLASH is in slot 3. Your floppy disk controller should reside in slot 4.

Once the controllers are firmly in place, turn on your Multi-Pak and then your Color Computer.

3. Using the FLASH

The innovative feature of the miniFLASH is its utilization of FLASH memory technology to store ROM code. This technology totally replaces the antiquated EPROM that is found on older floppy and hard drive controllers.

Unlike EPROMs, FLASH memory brings several benefits to Color Computer users. First, FLASH can be programmed in-circuit, without any additional hardware. Second, FLASH technology delivers a large amount of storage in an extremely small footprint. This allows multiple ROMs to be stored in the same chip and selected via hardware or software by the user.

How the FLASH is Partitioned

The miniFLASH contains 64K of internal FLASH, which is partitioned into four banks: bank 0, bank 1, bank 2 and bank 3. Each bank can hold 16K of code.

Typically, a ROM image will reside in each bank. These ROM images can be DOS's such as Disk BASIC, HDB-DOS or A-DOS, etc, or they can be ROM Paks, such as games and applications.

While all banks can hold code, only one bank is active at any given time. It is up to the user to determine which bank will hold code, and to which bank the interface will start from at power-on.

There are two methods of FLASH bank selection: the hardware switching method and the software switching method.

Hardware Switching

DIP Switches 2 and 3, located on the side of the miniFLASH, can select bank 0, 1, 2 or 3 as the desired bank to reference on power-up. Simply set up the switches for the desired bank to be active. Due to the location of the DIP Switch on the interface, we recommend that you turn OFF your system and remove the miniFLASH to properly set the switch.

If the desired FLASH bank contains a DOS such as Disk BASIC, HDB-DOS, A-DOS, etc., then the auto-start switch (DIP Switch 1) should be in the OFF (Up) position.

Conversely, if the desired FLASH bank contains a ROM Pak game image, you can elect to turn on the auto-start switch. This will cause the game to auto-start when your CoCo is powered up.

Software Switching

Use of the FLASH Bank Selection Register (FBSR) allows you to switch to a ROM in a specific FLASH bank without necessarily having to power down your CoCo. The procedure for software switching between a CoCo 1/2 and a CoCo 3 are slightly different, due to the fact that the CoCo 1/2 runs BASIC from ROM, whereas the CoCo 3 runs BASIC from RAM.

CoCo 1/2 FLASH Bank Software Switching

Depending on the type of ROM image, there are two different methods for software selection of a FLASH bank. If you are switching to a FLASH bank containing a DOS such as Disk BASIC, HDB-DOS, A-DOS, etc. the following commands will work on a CoCo 1/2:

```
POKE 113,0:POKE &HFF59,X (where X is the bank 0-3)
EXEC 40999
```

If the FLASH bank you are switching to contains a ROM Pak image such as a game, be sure the auto-switch (DIP Switch 1) is in the OFF (Up) position, and type the following on a CoCo 1/2:

```
POKE &HFF59,X (where X is the bank 0-3)
EXEC &HC000
```

Depending upon the contents of the target FLASH bank, your CoCo may go into a crash mode. This is normal, due to the fact that the ROM that BASIC is fetching instructions from is actually being totally replaced at the point of the POKE command. If this happens, press the RESET button and the code should execute properly.

CoCo 3 FLASH Bank Software Switching

Depending on the type of ROM image, there are two different methods for software selection of a FLASH bank. If you are switching to a FLASH bank containing a DOS such as Disk BASIC, HDB-DOS, A-DOS, etc. the following commands will work on a CoCo 3:

```
POKE &HFF59,X (where X is the bank 0-3)
POKE 113,0:DLOAD
```

If the FLASH bank you are switching to contains a ROM Pak image such as a game, be sure the auto-switch (DIP Switch 1) is in the OFF (Up) position, and type the following on a CoCo 3:

```
POKE &HFF59,X (where X is the bank 0-3)
EXEC &HE010
```

Determining the Active FLASH Bank

You can determine which of the four banks of FLASH memory is active by reading the FBSR. Type the following (assuming the miniFLASH's base address is $FF50):

```
PRINT PEEK(&HFF59)
```

4. Programming the FLASH

The miniFLASH Utilities disk image contains software to extract a ROM from a ROM Pak or disk controller. It also contains software to program a ROM image into any of the four FLASH banks on the interface.

Typically you will want to program the FLASH on the miniFLASH if you intend to use it in a specific application. In such a scenario, the interface would be plugged directly into the side of the cartridge port, with a DriveWire-compatible DOS such as HDB-DOS in one of the FLASH banks.

You may also want to just keep the miniFLASH in a slot in the Multi-Pak and have one or more ROM Paks programmed into the FLASH, allowing them to be selected on the fly.

Let’s take a look at different scenarios for programming the FLASH using the supplied utilities.

Programming an Existing ROM into FLASH

To program a ROM Pak or disk controller ROM into the FLASH, you have several options:

Option 1: Save a ROM Pak or disk controller EPROM’s contents to disk using GETROM.BAS, then program the desired FLASH bank in the miniFLASH using FLASH.BAS.

Option 2: Use ROMXFER.BAS to directly transfer the contents of a ROM Pak or disk controller’s EPROM to a FLASH bank (this requires a Multi-Pak).

For illustration, we will perform option #1.

Your first step is to choose the disk controller or ROM Pak that contains the ROM you want to transfer to FLASH. If it is a ROM Pak, **you will need to place tape over pin 8 to prevent the cartridge from auto-starting during the transfer process**. If it is a disk controller, then it isn’t necessary to cover pin 8.

In either case, once the cartridge is properly prepped, insert it into a free slot of the Multi-Pak. Slot 1 is usually a good slot for this.

Put your miniFLASH in slot 3 of the Multi-Pak, and your floppy disk controller in slot 4. Be sure the slot selection switch is set to 4.

Mount the miniFLASH Utilities disk image into your DriveWire Server and type:

```
    RUN”GETROM”
```

Follow the prompts and supply a filename. The contents of that disk controller or ROM Pak will be saved to the filename.

When the program is complete, you will be returned to Disk BASIC. Now, it is time to run the program to transfer the ROM on disk into the FLASH on the interface.

To start, type:

```
    RUN”FLASH”
```

The program will prompt you for the following information:

1. The base address of the miniFLASH
2. The MPI slot of the miniFLASH
3. The FLASH bank where the ROM image will be programmed
4. The name of the file to program

Once the information is provided, the file will be opened and its contents transferred to the appropriate bank of FLASH.

That’s all there is to programming the FLASH!

5. Supplied Software

The supplied miniFLASH Utilities disk image provides additional software which can assist you in setting up your interface and drives properly. We highly recommend that you work from a backup of your original disk.

All BASIC programs are saved in ASCII format, and therefore will load somewhat slower from the disk. To speed up subsequent loads, LOAD, then SAVE the program back to disk.

FLASH.ASM/FLASH.BIN

These two files make up the FLASH programming routine used by the included BASIC programs. The interface to the routine is documented later.

FLASH.BAS

This program uses FLASH.BIN and allows you to program the contents of a disk file into a FLASH bank of your choosing.

GETROM.BAS

This program uses the assembly routines in FLASH.BIN and allows you to save a ROM image from a controller or ROM Pak to disk. When run, GETROM.BAS will prompt you for the Multi-Pak slot where the ROM resides and the size of the ROM. It will then obtain the ROM's contents and save it to the disk file specified.

If this program is run on a CoCo 3, then additional options are available to save a 32K ROM Cartridge, or even the internal 32K ROM.

REMEMBER: For auto-start cartridges such as ROM Paks, be sure to place a piece of tape over pin 8 so that the cartridge does not auto-start. Also insure that DIP Switch 1 of the miniFLASH is in the OFF (Up) position. Failure to do so will result in a crash when the program starts the transfer!

ROMXFER.BAS

Like GETROM.BAS, this program also uses FLASH.BIN. The convenience of ROMXFER is that it will transfer the contents of a ROM Pak or disk controller's EPROM in one Multi-Pak slot to the desired FLASH bank in another Multi-Pak slot.

CHECKSUM.BAS

This program will provide a checksum on a ROM or bank of FLASH, returning the sum of all bytes. A blank FLASH bank should return the checksum value of 0, indicating that the FLASH contains all zeros.

ERASE.BAS

This program will erase a bank of FLASH by writing all $00s to the specified bank.

6. Technical Information

This chapter provides detailed technical information on the miniFLASH, including register layout and the software interface for programming the FLASH.

Hardware Register Layout

The hardware registers are documented below.

FLASH Bank Selection Register

This register is located in memory-mapped I/O and is settable to $FF49, $FF59, $FF69, or $FF79 via the jumper block on the board.

Bit 7 of this register is the FLASH Enable bit. This bit is set by the FLASH programming routine when writing to the FLASH. Unless you fully understand what you are doing, you should never set this bit.

Bits 6-2 are undefined. Reads of these bits will always return zero. Writes to these bits are ignored.

Bits 1-0 are used to software select one of the 16K FLASH banks (0 - 3).

FLASH Software Interface

Assembly language programmers who wish to access the FLASH read/write routines in FLASH.BIN may do so using the following information.

The FLASH.BIN routine loads at $7000 and can be called by JSRing to that location. The routine either reads or writes one page (128 bytes) of the FLASH. Subsequent calls are required to program an entire bank. In ROM Read mode, the contents of ROM Paks can also be read if the value at $7100 is $00.

There is a parameter block at $7100 that is referenced by the FLASH.BIN routine. That parameter block contains the following offsets:

Offset	Value
$7100	Read/Write Mode ($00 = Read ROM, $01 = Read FLASH, $FF = Write FLASH)
$7101	Multi-Pak Slot where FLASH or ROM is located (0-3, or $FF if no Multi-Pak is present)
$7102	FLASH bank to read/write (0-3)
$7103	Page to read/write (0-125 on a CoCo 1/2, 0-123 on a CoCo 3)
$7104	Page size (should be 128)
$7105-06	ROM base address (usually $C000)
$7107-08	Address of FLASH Bank Selection Register ($FF49, $FF59, $FF69 or $FF79)
$7109-0A	Address of a 128 byte buffer. This buffer either contains data to write to the FLASH (in write mode) or will hold the contents of a FLASH or ROM page (read mode).
$710B-0C	Two byte checksum of the contents of the FLASH/ROM page (in Read Mode).

Once the parameters from $7100 to $710A have been set, the caller may JSR into the subroutine at $7000 and the selected operation will be performed.

After reading a page of FLASH or ROM, a two byte checksum of that page is automatically computed and stored at $710B-0C. This value is the sum of all bytes in the page, and can be used as a verification mechanism.

Special Considerations

It is important to note that not all of the ROM can be read, nor all of the FLASH programmed. On a CoCo 2, all but the last 256 bytes ($FF00-$FFFF) of FLASH can be read/written. On a CoCo 3, all but the last 512 bytes ($FE00-$FFFF) of FLASH can be read/written.

The included BASIC software automatically detects the type of Color Computer being used, and makes the proper adjustments when reading/writing FLASH. Refer to the source code for consideration when writing your own FLASH programming software.

7. Additional Information

If Problems Arise

Although our products are engineered with quality and care, Cloud-9 cannot guarantee that you won't run into problems.

If you have a problem or question, please contact us via the Internet at support@cloud9tech.com. You can also visit our website at http://www.cloud9tech.com for up-to-date information on your product.

Manual Revision 1.1 – April 15, 2015

Cloud-9

Cloud-9
3749 County Road 30
Delano, MN 55328
Telephone 7639723261
www.cloud9tech.com