

MC998 Handheld FAQ

1. What is MC998 Handheld?

MC-998 is an advanced handheld battery powered smartcard reader. It features a Motorola Dragonball CPU with 1~4Mbyte of FLASH memory and 128~256kbyte of SRAM. Standard input device consists of a 16-key keyboard and a flying wheel, standard output device is a 128x64 dot matrix LCD panel with several icons. An RS-232 serial port and an IR port are default for serial communications, and an optional modem unit could be used as another serial communication device. Up to 3 card slots could be used (depending on card types). An optional RF unit could be used to receive paging information.

2. What cards can be read by MC998 handheld?

MC998 supports both memory cards and processor ones, there is no limit of the card types supported.

3. How many smart card interface units are built for MC998 handheld?

There are 2 full-functional smart card interface unit in MC998. One is the main interface, where card insert/gone events are detected. Any kind of smart card could be used in the slot connected to the main interface. The second one is the auxiliary interface, which could be configured as either a full-functional slot where all kinds of cards could be used, or 2 slots with reduced function. When the auxiliary interface is configured as 2 slots, there are 2 hardware options: 2 slots that only one card could be inserted into one of the 2 slots any time, while it' s full-functional; or 2 slots that 2 cards could be inserted simultaneously, while only CPU-based smart cards could be used. If it' s configured as the later, different card drivers should be used.

4. What Serial Communication Devices does MC998 have?

3 peripherals are connected to the UART unit of MC998: a standard RS-232 interface, an infrared transceiver, and an optional modem. They cannot be used simultaneously. Various baud rates are supported (up to 115200 for RS-232 and modem, up to 3579MHz for infrared).

The RS-232 interface is used to connect MC998 to another device using RS-232, such as PC, another MC998, a modem, or a printer, using the cable provided. The infrared transceiver could be used to connect to other IR devices. IrDA support is not a part of API, but it could be implemented using API. A simple 2400bps modem could be mounted so MC-998 could connect to phone line via the RJ11 socket directly, and additional device driver is required.

5. Does MC998 support a printer?

Yes, MC998 supports any printers that can do the communication via RS-232.

6. Does MC998 have a built-in Pager?

Yes, MC998 has an optional built-in paging information receiver. And only POCSAG pager is supported now, but FLEX paging will be available soon.

Attention: This option will increase the power consumption significantly.

7. How much system memory does MC998 have?

System memory of MC-998 consists of 128k~256k bytes SRAM and 1M~4M bytes FLASH ROM.

The SRAM is backup by a backup battery, the data stored could retain for several hours after the main battery has gone. Stack, heap, program data is stored here.

The FLASH ROM contains program code (including BIOS, configuration, user applications) and some program data (font data, constant strings, etc).

8. What is the difference between SRAM & FLASH ROM?

The main difference between FLASH and SRAM is that even though FLASH could be read just like SRAM, the writing is a quite different procedure, and modification is even more complicated: in the worst case, a block (up to 64k bytes) should be erased before any single bit writing could be performed. Since it's quite large, it could be used for non-violate data storage. A sample database driver using FLASH as data storage is provided.

9. How many FLASH ROM is left for my program?

It is differed on the variable size of FLASH ROM installed.

To 1MB FLASH ROM, 702KB space is left for user program. The start address is 0x00000000, the end address is 0x00100000.

To 2MB FLASH ROM, 1728KB space is left for user program. The start address is 0x00000000, the end address is 0x00200000.

To 4MB FLASH ROM, 3774KB space is left for user program. The start address is 0x00000000, the end address is 0x00400000.

10. Does MC998 have a Real Time Clock and Periodic Timer?

Yes, MC998 do have both of them. A real time clock (RTC) is built inside MC-998.

It's just like an ordinary watch, with date, time and alarm, which keep on running as long as the battery is there (no matter main or backup). It could be read/set by APIs, and sends a message when alarming.

Periodic timer is a single shot timer which could be set to 1/64~1024 second with 1/64s resolution.

Attention: Periodic timer will wake the system up every 1/64 seconds, so it will be more power consuming.

11. What is the software develop environment for the MC998 application?

GNU C is the main programming language, and the GNU m68k Tool Chain is used for developing. The whole GNU Tools Chain runs on a PC running MSDOS or an MSDOS Window under Win95/98.

12. What is GNU GCC Tools Chain?

The GNU GCC Tools Chain consists of C/C++ compiler, linker, assembler, and some binary utilities. All the tools have many UNIX-style command line options. It is suggested that a make file be used to simplify the whole compile-link process.

Attention: Please read the GCC manual for details.

13. Where can I get the GNU develop environment?

You can get the development environment for free by contact our sales.
Email: sales@magcard.com or call 1-408-4418858

14. How to start programming on MC998 Handheld?

Programming on MC998 is quite like programming on DOS, even simpler: there' s only one application that never returns. After the BIOS has done its job initializing the system, the control will be handed over to the application. The application is mainly written in C. In the C runtime routine (CRT0), i.e. start file, all variables are initialized. After that, it is the well-known main () subroutine. Please note that main () is a no-return subroutine.

15. What is a typical application flow to MC998 Handheld?

The main idea of MC998 application flow is called dead-loop. The application calls sysmsg () continuously to check for any events. After an event comes, the application would try to process it and then go back to sysmsg () for more. Actually there are two different ways to handle this in a multi-screen menu-based user interface:

One, The dead-loop itself is the message-polling loop (MPL). Each screen is registered with its own event dealers and screen descriptions/refreshers, so the MPL will know what to do with a certain event.

Two, each screen has its own MPL. Now the MPL does not have to be a dead loop. The second one is simpler but when the menu system gets larger, it will become more like a mess.

The first one is not a very good one for small scale projects, but it' s highly expandable and it' s easy to develop complicated GUIs with visual programming tools.

In fact, a menu driver could reside in the driver layer, and does not have to change from application to application at all.

16. Where can I get the MC998 Handheld API's document?

You can get the document either from the Magcard distributed MC998 SDK CD, or from our web site <http://www.magcard.com>

17. Can I download/upload database between PC & MC998 handheld?

Sure, you can. To archive this goal, you need to implement the stream read/write function code for both PC side and handset side. For example, if you want to send data via the standard RS-232 interface from PC to MC998, firstly you need to encode the desired data in a certain format output stream and send it to the standard RS-232 interface. Secondly, you need to tell MC998 how to enter receive mode, define & initialize the buffer for receiving stream. And trigger a event to end the receiving, decoding the input stream with the same format which has already been defined on PC side. Last you should invoke the DB related MC998 API to write all the data to MC998 FLASH ROM/Smart Card.

18. I need some samples on MC988 application programming, where can I get it?

You will find some basic samples from the Magcard distributed MC998 SDK CD. Also you can call our technical support dept for the further questions.

19. How to build the application binary code for MC998 Handheld?

Firstly, make sure all your application code has been saved in file (appmain.c). Secondly, execute batch file (m68path.bat). The syntax should be "m68path X:" -- X: represented by the local driver letter that installs your compiler utility. This file will set the system PATH parameter for you. Thirdly, make sure your (appmain.c) is in the same directory as the file (makefile). Finally, under the (appmain.c) directory, execute the command "make". If everything is OK, then you will get the downloadable binary file (appmain.bin).

20. What should I do if I get some error message when I build my application? Do you provide any Debugging tools?

You need to check your code carefully by following the instruction generated by the compiler. Debugging tools (debugger, simulator, etc.) is not provided but will be available later.

21. How to download applications to MC998 Handheld?

Please follow the steps below to download applications:

1. Turn on the Power of MC998
2. Connect MC998 to the PC's COM1 port via RS-232 cable provided.
3. Set the MC998 into Download mode
Hold the Power key (key code 0x00) while resetting. Inserting the battery will reset the system as well as pressing the reset key located in the back of MC-998, by the battery socket. The reset key could be pressed using a pin. A hint message ("downloader standing by") will be displayed on the screen if downloading mode is entered this way.
4. Execute the Updown.exe (which is contained in our SDK CD) application on PC, then the MC998 dialogue will appear on your PC
5. On the MC998 dialogue use the Browse button to select the *.bin file which you want to download
6. Press the Start button, you will get the downloading info from the main window
7. After download successful, MC998 will begin to execute the application automatically