FCI
Fossil Compatible Interface
Version 1.06 Oct 1994
Copyright 1994, Mario Viara
All Rights Reserved

Mario Viara
P.O BOX 130
12100 CUNEO ITALY

BBS: +39 171-691995
FAX: +39 171-691995
VOICE:+39 171-695128
NETMAIL: Mario Viara 2:334/705@fidonet.org
EMAIL: M.Viara@agora.stm.it

INDEX

1. INTRODUCTION 3

1.1. NOTICE 3

1.2. REGISTRATION INFORMATION 3

1.3. DISTRIBUTION NOTICE 3

1.4. Simple Description 4
1.4.1. System requirement 4
1.4.2. Most important properties 4
1.4.3. Contact the author 4
1.4.4. Distribution Form 4

2. INSTALLATION 5

2.1. Specific version 5

2.2. Multi task installation 5
2.2.1. Multi task performance 5

2.3. Command line parameter 5

3. CONFIGURATION 6

3.1. Configuration parameters 6
3.1.1. Generic parameters 6
3.1.2. Port specific parameters 6

4. ADVANCED CONFIGURATION 9

4.1. Detect idle time 9

4.2. Automatic Transmission/Reception 9
4.2.1. Automatic Transmission 9
4.2.2. Automatic Reception 9

4.3. FCIDUMP Utility 9

5. TECHNICAL INFORMATION 11

5.1. Tick functions and user function 11

5.2. Function CALL 11

5.3. FCI API 11

1. INTRODUCTION

1.1. NOTICE

THIS IS NOT FREE SOFTWARE! If you paid a "public domain" vendor for this program, you paid for the service of copying the program, and not for the program itself. Rest assured that nothing ever gets to the originators of this product from such a sale. You may evaluate this product for 30 days, after this evaluation period if you make use of it, you must register your copy.

1.2. REGISTRATION INFORMATION

The non-commercial single-user registration fee for FCI is $20 US.
The commercial and multisystem site fee for FCI is $50 US.
The check can be in US $ or equivalent value in other currency.
Send a check or company P.O. for the appropriate amount to:

Viara Mario
Casella Postale 130
12100 CUNEO - ITALY

After the registration you will receive one registration KEY and the instruction for register FCI and all new released version.
For more detailed information see the file REGISTER.TXT.

1.3. DISTRIBUTION NOTICE

This is "user-supported" software. You are hereby granted a license by Mario Viara to distribute this evaluation copy of FCI and its documentation, subject to the following conditions:

- FCI may be distributed freely without charge in evaluation form only.
- FCI may not be sold, licensed, or a fee charged for its use. If a fee is charged in connection with FCI it must cover the cost of copying or dissemination only. Such charges must be clearly identified as such by the originating party. Under no circumstances may the purchaser be given the impression that he is buying FCI itself.
- FCI must be presented as a complete unit, including this documentation. Neither FCI nor its documentation may be amended or altered in any way.
- By granting you the right to distribute the evaluation form of Con>Format, you do not become the owner of FCI in any form.
- Any other use, distribution or representation of FCI is expressly forbidden without the written consent of Mario Viara.

1.4. Simple Description

FCI is one implementation of FOSSIL 03 specification with some important and very interesting innovation.

1.4.1. System requirement

Fci required one system with all this properties :

- IBM-AT ISA or EISA Compatible architecture
- Intel 80286 or higher processor
- UART NS16450 or 16550 compatible

1.4.2. Most important properties

FCI have this important properties :

- 386 or higher specific version
- Can be loaded in UMB
- 32 different port
- Multi serial board support
- Multi tasking support
- Interrupt lost automatic recovery support
- Automatic support when lost one character of handshake

1.4.3. Contact the author

I'm very happy if you want send me your impression or hint for future version. For this purpose use one of mail or e-mail address present in the introduction page.

1.4.4. Distribution Form

FCI is distributed in one unique file called FCIxxxxb.ZIP that contains all documentation and executable file, xxxx is the version number and b if present is the identification for one beta version. For example the version 1.02 have the name FCI0102.ZIP.
The latest version of FCI can be download from my BBS with file request magic name FCI.

2. Installation

2.1. Specific version

FCI is distributed in two different form FCI.EXE and FCI386.EXE. FCI.EXE is a general version for all CPU from 80286 and FCI386.EXE is one special version that use some 386 specific instruction to improve performance and can be used in all system with one 80386 , 80486 or higher system. The two version are identical for all options only the 386 version is more fast. All instructions are for FCI.EXE but can be applied to FCI386.EXE.

2.2. Multi task installation

When work in one multi task environment FCI can be installed in two way ,in the common memory, if is installed before starting the multi tasking ,or in the task memory , if installed after the multi tasker. For multitasker supported like DesqView or Windows 3.x FCI must be, for better performance, before the multitasker for other multitaskter must be installed in any single task that use it.
One very important difference from FCI and other FOSSIL driver is that FCI detect and support automatically multi tasking and d'ont need TAME-aware program.

2.2.1. Multi task performance

In one multi task environment all time spent for wait character form serial board or for wait that one transmission is complete with one normal fossil driver is lost, FCI can automatically detect this time and give it to the system that can run other task, with greater improvement of general system performance.

2.3. Command line parameter

FCI for is installation take parameter from one file that default name is FCI.CFG and must be located in the current directory from where the driver is started. This filename can be changed using the F option described later.

For installing FCI use this command :

FCI [Options]

where options are :

- Filename - Can be used only for installing the driver and if is present filename override the default configuration file name FCI.CFG.
- I - Can be used only after installation and display some driver information.
- U - Can be used only after installation and if is possible uninstall the driver.

3. Configuration

Configuration of FCI is made by one ASCII text files that contain all parameter. This is file can be modified with any normal DOS text editor starting from the default FCI.CFG supplied with the packet. All line that start with one ; are comment and are ignored, comment can be put after any parameter.

3.1. Configuration parameters

The configuration file have this generic form :

NAME[x] [VALUE]

Where NAME is the name of the parameter that want change/set and VALUE if present is the relative value. NAME can be followed from one number for parameter that are port specific like base address , irq number etc. VALUE is not present for 'FLAG PARAMETER' that do not require one value but are only options that are activated.
In this document parameter that are not port specific are called generic parameter and the others port specific parameter.

3.1.1. Generic parameters

NAME	BASE	DEFAULT VALUE	DESCRIPTION
PORTN	10	4	Maximum number of ports that can be installed, the maximum current supported number is 32.
MAXWAIT	10	100	Number of nop call to the fossil driver before of automatic idle context switch.
MAXTWAIT	10	10	Maximum time in tick for MAXWAIT nop call.
MAXAUTORX	10	800	Time-out in tick after that if no chars are received the driver send one XON.
MAXAUTOTX	10	100	Maximum time in tick after one XOFF is received that driver enable automatically the transmission.

3.1.2. Port specific parameters

NAME	BASE	DEFAULT VALUE	DESCRIPTION
BASEx	16	3F8,2F8,3E8,2E8	Base address of the port.
PSTSx	16	-	Status port for multiserial board.
PMASKx	16	-	Status port mask for multiserial board.

PVALUEx	16	-	Status port value for multiserial board. When one interrupt from one multiserial board occurs the driver for decide the interested port read the PSTSx make the and with PMASKx and compare the value with PVALUEx if are the same the interrupt for the specified port is served otherwise skip to the next port configured in the same multiserial board.
IRQx	10	4,3,5,7	This is the interrupt request number for the port if is 0 the port is not interrupt driven but polled. Polled port can work fine with low speed baud rate in non multitasking environment.
RXSIZEx	10	1024	Size of recv buffer.
TXSIZEx	10	512	Size of tx buffer.
RXVHIx	10	RXSIZEx*7/8	Limit of number of char in recv buffer that when is reach if XON/XOFF is enabled the driver send one XOFF for every char received.
RXHIx	10	RXSIZEx*3/4	Limit of number of char in recv buffer that when is reach the driver, if enabled, send one XOFF and lower the CTS signal.
RXLOWx	10	RXSIZEx/2	Lower limit of handshake when if reach the driver ,if enabled, send one XON and rising the CTS signal.
BAUDx	10	-	If this parameter is specified the baud rate is locked to the specified value and every try to change it take no effect. All baud rate from 300 to 115200 can be written the only exception is for 115200 baud that must be written 11520.
FIFOx	10	8	Is the value of receive FIFO buffer for uart that support it. This value can be raised if the system have one very low interrupt latency.
FLAGIDLEx	FLAG	NO	If specified the driver detect idle time, for multitasking environment, according to MAXWAIT and MAXTWAIT and give it to the system.
FLAGAUTOTXx	FLAG	NO	If present the system enable the logic of auto transmission according with MAXAUTOTX.
FLAGAUTORXx	FLAG	NO	If present the system enable the logic of auto receive according with MAXAUTORX.

FLAGINT13x	FLAG	NO	If present when one INT 13 call is invoked automatically the system disable the RTS during the call. This flag is very useful for system that take for too much time the interrupt disabled during I/O.
FLAGDSRx	FLAG	NO	The driver return DSR signal every time active without regarding to the real state.
FLAGCTSx	FLAG	NO	The driver return CTS signal every time active without regarding to the real state.
FLAGDCDx	FLAG	NO	The driver return DCD signal every time active without regarding to the real state.

4. Advanced configuration

Some characteristic of FCI are very useful but need one more detailed explanation for use it in the most good way, this chapter explain of configure FCI in the most typical situation.

4.1. Detect idle time

MAXWAIT and MAXTWAIT are very important for improve performance in multitasking environment, this parameter permit to FCI of decide when must give control to the system.
When one fossil call is made without input/output transfer, for example check if some haracter are available or check if the transmission buffer is full, FCI understand that this is one 'nop' call. MAXWAIT is the number of consecutive 'nop' call must occur before that the control is give to the operating system. MAXTWAIT is the maximum time ,in system tick (each 55 ms), in that the MAXWAIT is checked.
FCI detect automatically all other condition that the idle time must release to the operating system, for example if one program wait for one character to be received and have no character in the receive buffer all the time that must wait for one character available is released to the operating system.
Detect idle time can be enable or disable for any port via the flag FLAGIDLEx.

See also : Port specific parameters

4.2. Automatic Transmission/Reception

FCI implement one logic of automatic transmission/reception that is very useful when one character of handshake is lost. This logic is controlled by two parameter MAXAUTOTX and MAXAUTORX.

4.2.1. Automatic Transmission

When FCI receive one XOFF for one specific port, if the relative FLAGAUTOTXx is enabled, start one counter that when reach MAXAUTOTX without receive one XON, FCI cancel the effect of the XOFF and start to send data. This function is very useful for bad quality line that lost many characters.

4.2.2. Automatic Reception

Automatic reception work like automatic transmission but in opposite front, if FCI detect that one port , with FLAGAUTORXx enabled, d'ont receive character for one time that is greater of MAXAUTORX send automatically one XON.

4.3. FCIDUMP Utility

With FCI is present one program named FCIDUMP.EXE. The scope of this program is monitoring FCI activity and is very important during debug fase but also I have include it in the distribution package because can also be used for configure FCI for

best performance. FCIDUMP run in two form. If is invoked without command line options send to the standard output all parameter of FCI. If invoked with any options display the same parameter in real time, this last form is very interesting if used in one multi task environment for example is possible monitoring FOSSIL activity of all task that running in the system. One complete explanation of all the parameter displayed is out of scope of this document but all the name of the parameter that are displayed are self explained and I think that any one that now how one fossil driver work can understand all.

5. Technical information

In this chapter I analyze how FCI handle some situation that are not completely clear in the FOSSIL specification.

5.1. Tick functions and user function

FCI like other fossil driver handle tick and user function, but multitask environment need some little clarification. If the function is connected before that multitasking is started this function is called at every tick but the function running in non specific V86 machine and cannot access to any process specific function. Otherwise if the function is connected after starting of multitasking the function is specific for the task the connect it and can access to the data of the specified V86 machine. Normally is better load program that user tick or user defined fossil functions in the environment of the process that use it.

5.2. Function CALL

All function not in the fossil specification, for example all function call with number greater of 1ch are passed without any check to the previous INT 14 handler.
All fossil function that reference one port not open are again passed to the previous INT 14 handler, with exclusion of fossil open port function.

5.3. FCI API

Over of fossil specification FCI implement one internal API any one that is interested can ask the author for one copy of FCI API specification.