PROCESS AND ITS DEVICE TO CONNECT ON THE SAME SERIAL PORT ALREADY USED, ANOTHER PERIPHERAL

The invention has for its object a process and its device to connect on the same serial port of a computer already used with a peripheral, another peripheral.

The object is to connect a peripheral such as a smart card reader or any other peripheral between a computer and a serial port already used and connected with any other peripheral (for example a mouse, a joystick, a modem, ...). This serial port uses a communication protocol of the type RS232 or I2C or SPI or the like, ...

According to the state of the art, if we take for example a mouse, the latter is connected directly by a series connection. To carry out data exchange between the mouse and the computer, there is used a "mouse driver" and a "UART" in the computer (universal asynchronous emitter/receiver) which serves as a means for communication management.

It is important to note that according to the state of the art, if the user wishes to use a mouse on his computer, he is obliged to configure the input/output port used by the mouse. Because of this, the user can use only the mouse on this port, and no other peripheral can be used on this same serial port.

The invention seeks to overcome this problem.

The state of the art can be defined by the following patent:

- FR-A-2.740.885: a process for the management and execution of security applications in interactive electronic equipment provided with a peripheral apparatus comprising secured means and pointing means, comprises an extended communication protocol of the communication protocol specific to the pointing means of said peripheral apparatus. This protocol permits the use of a single serial port for communication between the application software and the pointing means and the security means associated in the peripheral apparatus according to the invention. The peripheral apparatus integrates its pointing and terminal functions for microcircuit cards. Specifically to this invention, it comprises secured means for the storage, management and execution of the security applications. The process and the apparatus according to the invention are adapted for applications for the protection of software and management of user licenses of software particularly on interactive electronic equipment.

The process according to the invention, to connect to the same serial port of a computer already used by a peripheral (A), another peripheral (B), consists in:

- deactivating the drive of application (A), to be able then to proceed to the transfer of data without a problem for the cursor controlled by application (A),

- creating a new drive to control the drive of application (A) and the serial port of the computer,

- directly connecting the transmission channel to the universal asynchronous emitter/receiver (UART) of application (B) and application (A), said channel being used solely by the drive of application (A) to detect the material at start-up, said channel is also available at any time for application (B),

- in that the UART of the smart card reader cuts application (A) as soon as it receives data and immediately sends a response to the computer,

- if the computer is not itself requested for data, the UART of the card reader verifies that the request for data has been received and so informs the computer, then the computer applies an interrogation scenario, during this scenario the computer is master and the application (B) is slave.

The interrogation scenario is used so as not to disturb the user of the mouse during the time of response of the card reader.

To this end, said scenario is characterized by three parts:

→ sending a principal data request to the card reader, then in certain cases the card reader can send information to confirm good receipt of the requested, or directly the requested data,

→ use of interrogation to know whether the smart card reader has received the data from the card and if it is ready to send the principal data request,

→ searching for the data in response to the principal request.

According to another process of embodiment, a supplemental feed contact by the signals "request for invitation to send" (RTS) and the "ready to receive" (DSR) from a computer with a serial port, said contact directly connects the mouse to the computer in the case of cutting the supply of the smart card (or other peripheral).

The invention also provides the device for practicing the process.

The device to connect on the same serial port of a computer already used by a peripheral (A), another peripheral (B), is characterized by the fact that it comprises:

* in the computer:

- a new "driver", "other software interface" which serves as a "principal driver" connected on the one hand to the peripheral driver A (already used on said serial port) and which serves as a "slave driver" and on the other hand to the physical interface ($UART_1$) which serves as a communication management means.

* in the new interface of the other peripheral (B):

- a controlled switch

- an interface ($UART_2$) serving as a means for communication, these two elements being interconnected and being also connected to the physical interface ($UART_1$) of the computer to the peripheral (A) already used.

According to one embodiment, the peripheral (A) already used on the serial port of the computer is a mouse and the other peripheral (B) is at least one smart card reader.

The accompanying drawings are given by way of indicative but non-limiting example. They show an embodiment according to the invention. They will permit easy comprehension of the invention.

Figure 1 is a view representing the actual architecture.

Figure 2 is a schematic simplified view of the device according to the invention.

Figure 3 is a view according to Figure 2, showing another mode of presentation.

Figure 4 is a view of the scenario according to the process. In the course of this scenario, the user can work normally with the mouse without problem.

DESCRIPTION AND ARCHITECTURE

To transmit data from the computer 6 to the smart card reader 8 by the same serial port as the mouse 9, it is necessary to deactivate the drive of the mouse 3, to be able then to proceed to the transfer of data without problem for the cursor controlled by the mouse.

It is necessary to create a new drive permitting controlling the drive of the mouse 3 and the serial port of the computer.

The transmission channel is connected directly to the universal asynchronous emitter/receiver (UART) of the

smart card reader 7 and to the mouse 9, this channel is used only by the mouse drive to detect the mouse material at start-up of the user system, it is also available at any time for the smart card application.

When it receives data, the UART of the card reader cuts the mouse and immediately sends a response to the computer. Thus, the user of the mouse is not troubled by this discrete transmission of data.

If the computer 6 is not interrogated as to data, the UART of the card reader 7 verifies that the request for data has been received and so informs the computer 6, then the computer 6 applies an interrogation scenario until the smart card reader 8 receives a request for data. This interrogation scenario is described above in the description and is shown in Figure 4.

Throughout this process, the computer 6 is the master and the UART of the card reader 7 is the slave.

Figure 3 shows the architecture used:

1/ Application CARD READER

2/ User system

3/ MOUSE drive

4/ CARD READER drive (Driver of the other interface - software interface)

5/ Universal asynchronous emitter/receiver UART 8250 - SERIAL PORT

6/ COMPUTER

7/ UART interface of the CARD READER

8/ MATERIAL of the CARD READER

9/ MOUSE

10/ Control switch

The interrogation scenario is the following:

The time of response of the smart card reader 8 can be several minutes and the mouse 9 cannot remain out of service during all this time. Also, it is necessary to use a scenario so as not to disturb the user of the mouse 9 during the response time of the card reader 8.

The scenario is divided into three phases:

→ In the first phase, sending a principal data request to the smart card reader 8, then the card reader 8 sends information to confirm good reception of the request.

→ In the second phase, use of the interrogation process according to the scenario described to learn whether the smart card reader 8 has received data from the card (not shown) and whether it is ready to send the principal data request.

→ In the second phase, search of the data before response to the principal request.

The scenario is shown in Figure 4:

1/ Principal data request

2/ OK response

3/ Phase 1

4/ OK data request

5/ OK response

6/ Phase 2

7/ Data request

8/ Phase 3

9/ Principal data response

→ In the course of this scenario, the user can work normally with the mouse 9 with no problem.

It should be noted that the mouse 9 operates in the absence of supply of the smart card reader 8.

Another embodiment comprises another contact supplied with signals of "request for invitation to send" (RTS) and the "ready to receive" (DSR) from a serial port computer. This contact connects directly the mouse to the computer in the case of supply cutoff.

The serial mouse uses a data package of fixed format at a speed of 1200 bauds toward the host in the case of pushing on a button or detection of a movement of the mouse.

→ Initializing Reset

The mouse transmits a character "M" as soon as it is powered or in case a Reset is initiated by the host. This character informs the host of the conformity of the message to the constructor serial format.

→ Format of the message of serial data

During transmission of data to the host, the mouse defuses the following data format, according to the constructor serial interface. The format of data 1200 bauds uses a data block of 9 bits constituted by 7 bits of data message and two bits of control.

The mouse controller transmits the following data package of three octets, according to the interface, in response to a mouse event.

1/ Octet 1 of message data

Bit	Designation
1	Starting bit (always 0)
2	Bit 6 of horizontal movement
3	Bit 7 of horizontal movement (strongly weighted bit)
4	Bit 6 of vertical movement
5	Bit 7 of vertical movement (strongly weighted bit)
6	1 = Secondary button depressed
7	1 = Primary button depressed
8	Always = 1 (message synchronization bit)
9	Stop bit (always 1)

2/ Octet 2 of message data

Bit	Designation
1	Starting bit (always 0)
2	Bit 0 of horizontal movement
3	Bit 1 of horizontal movement
4	Bit 2 of horizontal movement
5	Bit 3 of horizontal movement
6	Bit 4 of horizontal movement
7	Bit 5 of horizontal movement
8	Always = 0
9	Stop bit (always 1)

3/ Octet 3 of message data

Bit	Designation
1	Starting bit (always 0)
2	Bit 0 of vertical movement
3	Bit 1 of vertical movement
4	Bit 2 of vertical movement
5	Bit 3 of vertical movement
6	Bit 4 of vertical movement
7	Bit 5 of vertical movement
8	Always = 0
9	Stop bit (always 1)

UART 8250

This UART (universal asynchronous emitter/receiver) permits controlling the serial port interface.

The UART 16550 is compatible, at the software and pinning level, with the 8250 but has an internal latency file operating on the principle of "first in, first out" (FIFO) and being adapted to be activated and deactivated by the software.

The COM1 port normally uses the address 03F8H, COM2 the address 02F8H, COM3 the address 03E8H and COM4 the address 02E8H. This lowest address is generally called the base port address and each addressable register is then referenced by relative addressing from this base value, as shown by this table:

If the bit DLAB of the LCR (communication control register) = 0:

00H DATA In case of reading, register of received data. In case of writing, register of holding transmission (reading/writing).

01H IER Register of validation of interruption (reading/writing).

If the bit DLAB of the LCR = 1:

00H Baud0 Swing divider BRG, low octet (reading/writing)

01H Baud1 Swing divider BRG, high octet (reading/writing)

Not affected by the bit DLAB:

02H IID Interruption identifying register (reading) and control register FIFO (writing)

03H LCR Communication control register (reading/writing)

04H MCR Modem control register (reading/writing)

05H LSR Communication condition register (reading)

06H MSR Modem condition register (reading)

Port 3F8/3F9 - Table of transmission speed divider (reading/writing) Bit DLAB in LCR = 1

BITS/SECOND	DIVIDER	BITS/SECOND	DIVIDER
50	2304	1800	64
75	1536	2000	58
110	1047	2400	48
134	857	4800	24
150	768	9600	12
300	384	19200	6

600	192	38400	3
1200	96	57600	2
1800	64	115200	1
2000	58		

Divider = 1843200 / (speed of transmission * 16).

Port 3F8 - Switching validation register IER (reading/writing) DLAB in LCR = 0

Data register received in case of reading.

Register for holding transmission in case of writing.

Port 3F9 - Register of switching validation - IER (reading/writing) bit DLAB in LCR = 0

Bit	Name	Binary	Meaning
7,6,5,4	Reserved	0000 xxxx	Reserved (zero)
3	EMI	xxxx 1xxx	Validation of switching in the modem condition
2	ELI	xxxx 1xxx	Validation of switching in the communication condition
1	ETI	xxxx xx1x	Validation of switching of the transmission register
0	ERI	xxxx xxx1	Validation of ready received data switching

Port 3FA - Register of switching identification - IIR (read only)

Bit	Name	Binary	Meaning
7,6	FIQ	11xx xxxx	16550 only, the holding files FIFO are validated.
		00xx xxxx	8250 and 16550, the holding files FIFO are devalidated.
5,4	Reserved	xx00 xxxx	Reserved (zero)
3	TINT	xxxx x1xx	16550 only, int. of session output. Holding.
		xxxx x00x	8250 and 16550
1,2	SINT	xxxx x00x	Interruption of modem condition, bit modified in MSR
		xxxx x01x	Interruption of transmission register, holding register of
		xxxx x10x	transmission empty, bit at 1 in LSR

		xxxx x11x	Interruption of ready received data, date of register full, bit at 1 in LSR. Interruption of communication condition, BREAK or bit of error in LSR.
0	IP	xxxx xxx1	Interruption in holding (no active interruption)
		xxxx xxx0	No interruption in holding

Port 3FA - 16550 Register for control FIFO - FCR (read only)

Bit	Name	Binary	Meaning
7,6	TL	00xx xxxx	Level of triggering = 1 octet to interrupt FIFO receiver
		01xx xxxx	Level of triggering = 4 octets to interrupt FIFO receiver

			10xx xxxx	Level of triggering = 8 octets to interrupt FIFO receiver
			11xx xxxx	Level of triggering = 14 octets for interrupt FIFO receiver
	5,4	Reserved	xx00 xxxx	Reserved (zero)
	3	Reserved	xxxx 1xxx	Modification of the pins RXRDT & TXRDY of mode 1 in mode 1
	2	CLT	xxxx x1xx	Emptying FIFO emitter
	1	CLR	xxxx xx1x	Emptying FIFO receiver
	0	IP	xxxx xxx1	Validation of emptying the emitter and receiver FIFO files

Port 3FB - Register of communication control - LCR (reading/writing)

Bit	Name	Binary	Meaning
7	DLAB	0xxx xxxx	Relative address to 0 returned to DATA, relative address 1 to IER

	control	1xxx xxxx	Relative address 0 and 1 returned to the divider swing BR of address
6	SETBRK	x0xx xxxx	Normal UART operation
	control	x1xx xxxx	Send interrupt signal
5,4,3	GENPAR	xxxx 0xxx	No parity bit
	control	xx00 1xxx	Parity bit = uneven
	of parity	xx01 1xxx	Parity bit = pair
		xx10 1xxx	Parity bit = 1
		xx11 1xxx	Parity bit = 0
2	XSTOP	xxxx x0xx	1 only stop bit
	stop bit	xxxx x1xx	2 stop bits (1,5 if WL = 5)
1,0	WD5	xxxx xx00	Word length = 5
	WD6	xxxx xx01	Word length = 6
	WD7	xxxx xx10	Word length = 7
	WD8	xxxx xx11	Word length = 8

Port 3FC - Control register of MCR modem (reading/writing)

Bit	Name	Binary	Meaning
7,6,5	Reserved	000x xxxx	Reserved (zero)

4	TEST	xxx1 xxxx	Activation of the self-test configuration UART
3	OUT2	xxxx 1xxx	Control of interruption signals 8250 (User2 output)
2	OUT1	xxxx x1xx	Restart of the internal modem 1200 b Hayes (User1 output)
1	RTS	xxxx xx1x	Pass to output RTS 1 toward the connector RS232C
0	DTR	xxxx xxx1	Pass to the DTR output 1 toward the connector RS232C

Port 3FD - Register of communication condition - LSR (read only)

Bit	Name	Binary	Meaning
7	PE/FE/int-err	1xxx xxxx	Interruption PE/FE 16550 in the holding file FIFO
		0xxx xxxx	Use of 8250 or 16450
6	TSRE	x1xx xxxx	Register of empty transmission offset

5	THRE	xx1x xxxx	Register of holding empty transmission
4	BI	xxx1 xxxx	Interruption received
3	FE	xxxx 1xxx	Frame error
2	PE	xxxx x1xx	Parity error
1	OE	xxxx xx1x	Overshoot error
0	DR	xxxx xxx1	Ready received data

Port 3FE - Register of modem condition - MSR (read only)

Bit	Name	Binary	Meaning
7	DCD	1xxx xxxx	Level detected of data carrier
6	RI	x1xx xxxx	Level of call indicator
5	DSR	xx1x xxxx	"Ready to receive" level
4	CTS	xxx1 xxxx	"Ready to transmit" level
3	DCDC	xxxx 1xxx	Change of DCD
2	RIC	xxxx x1xx	Change of RI
1	DSRC	xxxx xx1x	Change of DSR
0	CTSC	xxxx xxx1	Change of CTS

REFERENCES

1. CARD READER application
2. System of exploitation
3. MOUSE drive
4. CARD READER drive (Other interface driver - software interface)
5. Universal asynchronous emitter/receiver UART 8250 - SERIAL PORT
6. COMPUTER
7. UART interface of the CARD READER
8. CARD READER HARDWARE
9. MOUSE
10. Controlled switch

CLAIMS

1. Process to connect to the same serial port of a computer (6) already used by a peripheral (A), another peripheral (B), characterized by the fact

that it consists in:

- deactivating the application drive (A), so as then to be able to transfer data without a problem for the cursor controlled by the application (A),

- creating a new drive to control the drive of the application (A) and the serial port of the computer (6),

- directly connecting the transmission channel to the universal asynchronous emitter/receiver UART (5) of the application (B) and the application (A), said channel being used only by the drive of the application (A) to detect the start-up hardware, said channel being also available at any time for the application (B),

- the UART of the card reader (7) cutting the application (A) as soon as it receives data and sending immediately a response to the computer (6),

- if the computer (6) has not requested data, the UART of the card reader (7) verifies that the data request has been received and so informs the computer (6), then the

This Sheet Has Been Replaced

* C L M S *

computer applies an interrogation scenario, during which the computer (6) is the master and the application (B) is the slave.

2. Process according to claim 1, characterized by the fact

that said scenario is characterized by three parts:

→ sending a principal data request to the card reader (1), then the card reader (1) sends information to confirm the good reception of the request,

→ use of interrogations to determine whether the smart card reader (1) has received data from the card and if it is ready to send the principal data demand,

→ searching the data in response to the principal demand.

3. Process according to claim 1, characterized by the fact

that a supplemental contact supplied by signals "request for information to send" (RTS) and "ready to receive" (DSR) from a computer with a serial port, connected directly to the mouse (9) of the computer (6) in the case of cutting the supply of the smart card reader (1) (or other peripheral).

4. Device for practicing the process, according to claim 1, to connect on the same serial port of a computer already used by a peripheral (A), another peripheral (B), characterized by the fact

that it comprises

This Sheet Has Been Replaced

* C L M S *

* at the level of the computer:

a new "driver", another software interface which serves as "principal driver (4)" connected on the one hand to the driver of the peripheral (A) (already used on said serial port) and which serves as a "slave driver (3)" and on the other hand to the physical interface ($UART_1$) (7) which serves as a means of communication management

* at the level of the new interface of the other peripheral (B):

♦ a controlled switch (10)

♦ an interface ($UART_2$) (7) serving as a means for communicating these two elements being connected to each other and being also connected to the physical interface ($UART_1$) (5) of the computer (6) to the peripheral (A) already used.

5. Device according to claim 1, characterized in that

the peripheral (A) already used on the serial port of the computer (6) is a mouse (9).

6. Device according to claim 1, characterized by the fact

that the other peripheral (B) is a smart card reader (1).

This Sheet Has Been Replaced