Software Development Seminar

Peripherals (Basic)

Peripherals

Peripherals

Peripherals

1. Controller

    Multiple controller types can be used

2. Kanji (Chinese character) fonts

3. Memory card

    Rules governing hardware use and operation

    Error processing

    Test for presence

    Conflicts with controllers

    Examples of event-driven presence tests

Controllers
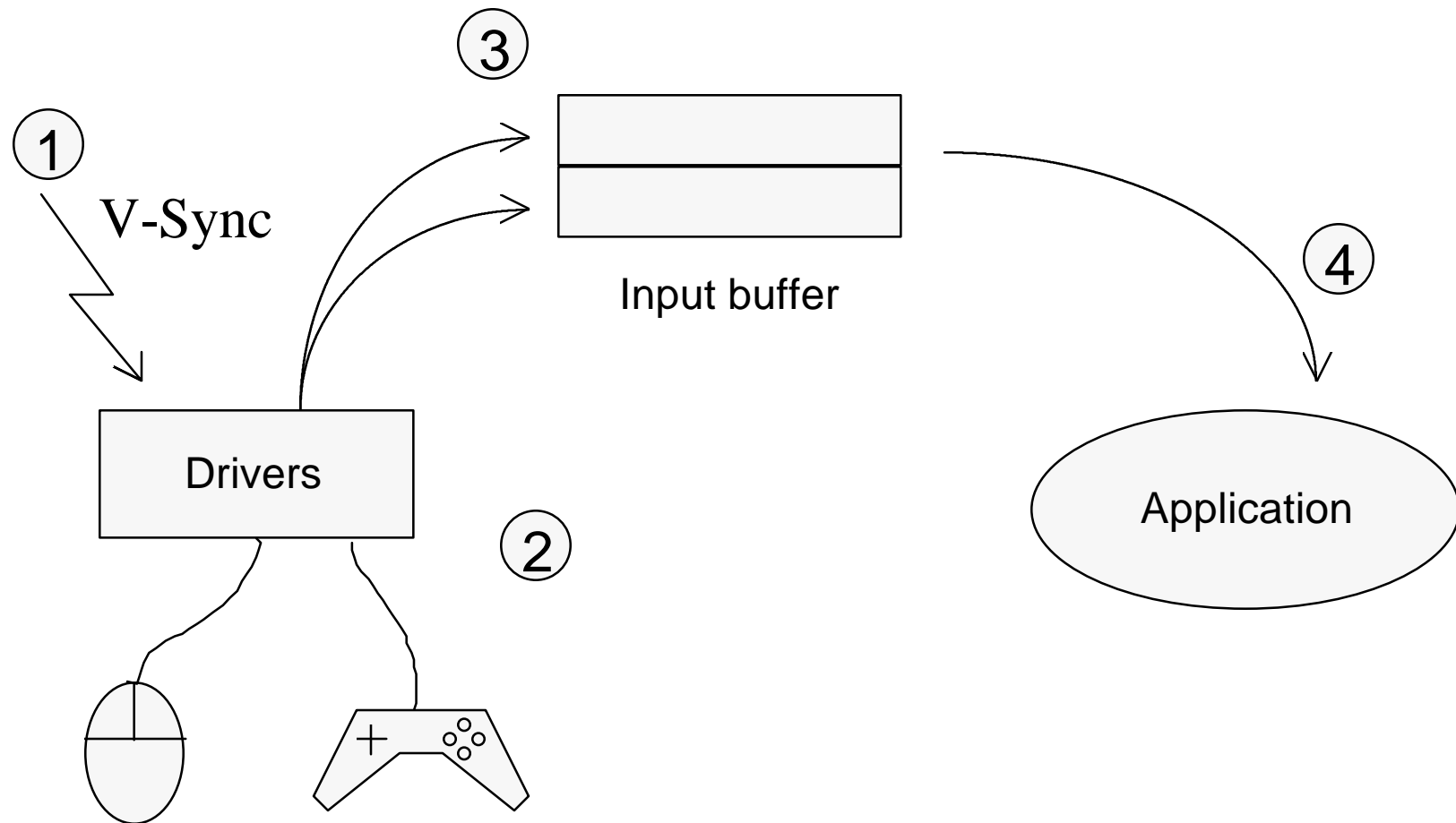
Controller access mechanism

③

① V-Sync

Input buffer

④

Drivers

②

Application

Initialization of controller BIOS

```
/* Input buffer */
unsigned char buf0[len0], buf1[len1];

InitPAD(buf0,len0,buf1,len1);
StartPAD();
```

CONFIDENTI
AL

Input buffer data format (1/2)

```
===================================================
Byte        Contents

-------------------------------------------------------------------------------
0           Input result        0x00 : successful    0xff : failed
1           Upper four bits : terminal type
              0x1 : mouse
              0x2 : 16 button analog A
              0x3 : gunì
              0x4 : 16 button
              0x5 : 16 button analog B
              0x8 : multitap
            Lower four bits: input byte count / 2


<16 button>
2,3         State of button     1 : Released       0 : Pressed
-------------------------------------------------------------------------------
```

CONFIDENTIAL

```
=================================================
Byte        Contents
-------------------------------------------------------------------------


<Mouse>
2                        unused
3           Button states         1 :  released        0 : pressed
                            bit 2: right            bit 3: left
4           Displacement along X-axis : -128 - 127
5           Displacement along Y-axis : -128 - 127
6           Displacement along Z-axis : -128 - 127
7,-         Displacement along ?    : -128 - 127


<16 button analog A, B>
2,3         Button state          1: released          0: pressed
4,5,6,7,-   Analog channel value


<Gun>
2,3,-       Button data
            Cursor position is stored in another buffer
-------------------------------------------------------------------------
```
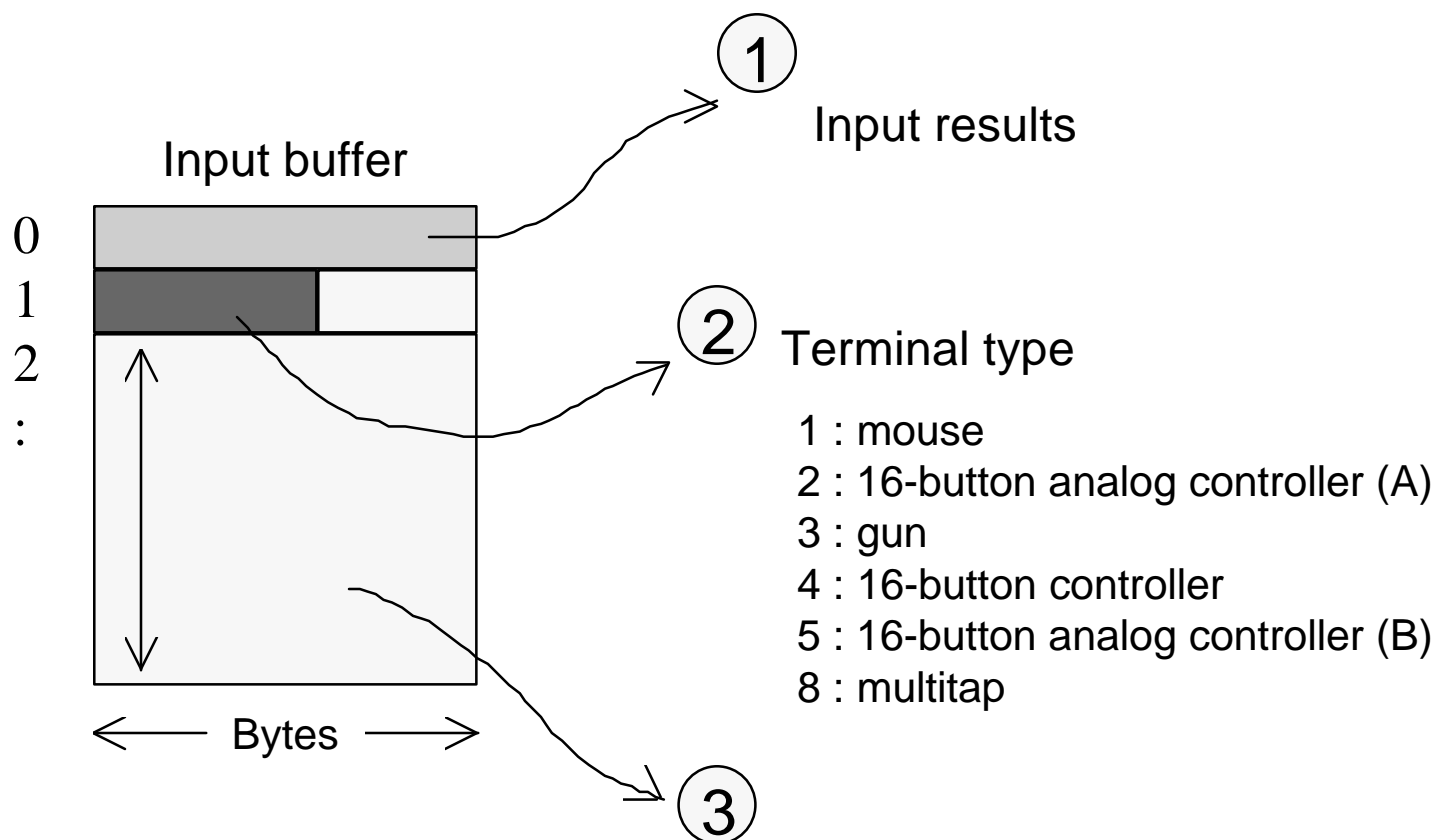
Multitap input data

```
====================================================================
 Byte        Contents
--------------------------------------------------------------------
0            Input results        0 : successful        0xf f : failed
1            0x80 (fixed)
2            Connector #1         Input results         0 : successful         0xff : failed
3            Connector #1         (Terminal type << 4)       |        (input byte count / 2)
4-9          Connector #1         Input data
10           Connector #2         Input results         0 : successful         0xff : failed
11           Connector #2         (Terminal type << 4)       |        (input byte count / 2)
12-17        Connector #2         Input data
18           Connector #3         Input results         0 : successful         0xff : failed
19           Connector #3         (Terminal type << 4)       |        (input byte count / 2)
 20-25       Connector #3         Input data
 26          Connector #4         Input results         0 : successful         0xff : failed
 27          Connector #4         (Terminal type << 4)       |        (input byte count / 2)
 28-33       Connector #4         Input data
--------------------------------------------------------------------
```

Automatic controller recognition

Input buffer

```
0
1
2
:
```

← Bytes →

① Input results

② Terminal type

    1 : mouse
    2 : 16-button analog controller (A)
    3 : gun
    4 : 16-button controller
    5 : 16-button analog controller (B)
    8 : multitap

③

ROM Kanji font

ROM Kanji font (cont)

```
=================================
Contents          File
-------------------------------------------------------

Library           libapi.lib
-------------------------------------------------------
```

* Actual patterns are in the boot ROM

CONFIDENTIAL

Fonts

Data format   16 dots x 16 dots. 2-value bit map.
Character size is 15 dots x 15 dots.

Contents   JIS level 1 kanji and non-kanji characters. The gothic
non-kanji characters include the top space (0x2121).

Access method   A shift-JIS code is provided to the service function,
which returns the header address of the font pattern
for the specific character in ROM. From there, the
font pattern can be directly accessed.

Data format

The upper left byte of the pattern is the header. The upper right byte follows. Bits are organized with the MSB to the left.

# 0	# 1
# 2	# 3
. . .	. . .
. . .	. . .
. . .	. . .
# 30	# 31

Related function

Krom2RawAdd   Get kanji font pattern address

Memory card

CONFIDENTIAL

Memory card library

```
========================

Contents        Filename
--------------------------------------

Library         libcard.lib
Header          kernel.h
                sys\file.h

--------------------------------------
```

Hardware

===

Capacity   120 Kbytes formatted (accessed in units of 128-byte
                  sectors) : Asynchronous serial
                  communication through the controller port

Communication   A synchronous serial communication through
protocol   the controller port

Access speed   (1) No access for 20msec after writing 1 sector
                  (2) Maximum continuous reading speed is
                      approximately 10Kbytes/sec

Other   No battery required
                  Can be inserted or removed without turning off the
                  power supply
                  Guaranteed for 100,000 write operations

BIOS

```
===============================================================
```

Access   128 bytes per 2 VSyncs

Activation timing   After VSync interrupt, a controller read is performed,
                    card presence is detected and handshaking is performed.
                    Data transmission and reception is performed with receive
                    interrupts for each byte.

Execution speed   30 sectors/sec = 3.75KB/sec

CPU load   continuous reads from two cards: 2.5 percent
                    continuous writes from two cards: 3.2 percent

```
---------------------------------------------------------------
```

File system

```
================================================================
```

Device name   buX0, X: connector number (0 or 1)

Filename   Up to 21 ASCIZ characters

Directory structure   None

Management unit: slot   8KB (64 sectors) ➤ file size unit

Number of slots   15 per card (Maximum file number is 15

File size   Specified in CREATE. Fixed thereafter.

Block   Required memory capacity as shown in the
          product catalog
          Same as "slot" (8KB)

```
-----------------------------------------------------------------
```

Function table <File system>

<File system>

open	(refer to directory cache)
read	(non-synchronous type)
write	(non-synchronous type)
close	(no card access)
firstfile	(refer to directory cache)
nextfile	(no card access)
delete	(refer to directory cache)
rename	(refer to directory cache)
format	(approx 1.2 sec complete CPU monopoly)
lseek	(no card access)

Known bugs (partial list)

- When a file is created using open(), the file should immediately be closed by calling close(). Otherwise, issuing read() or write() will generate an error.

- Asynchronous access using read() and write() will result in the file pointer being updated with 128 fewer bytes. Correction must be made with lseek().

Function table <BIOS>

<BIOS>

InitCARD	initialize memory card BIOS
StartCARD	activate memory card BIOS
StopCARD	stop memory card BIOS
_bu_init	initialize memory card file system
_card_info	get card state
_card_clear	clear unconfirmed flags
_card_load	test logical formatting
_card_auto	define settings for automatic formatting
_new_card	change settings for unconfirmed flag test
_card_status	get memory card BIOS status
_card_wait	wait for completion of processing by memory card BIOS
_card_chan	get memory card BIOS event
_card_write	write to one block in memory card
_card_read	read from one block in memory card

Initialization sequence

Cases when using simultaneously with controller
    InitPAD(&cbuf[0][0],34,&cbuf[1][0],34);
    StartPAD();
    InitCARD(1);
    StartCARD();
    _bu_init();

Cases when using simultaneously with other libraries
    (1) Initialization of ResetCallback() on other library
        containing it
    (2) Same sequence as above
    (3) ChangeClearPAD(0);

CONFIDENTIAL

Connected events

```
===============================================================
Descriptor        Classification      Meaning
---------------------------------------------------------------
 HwCAR            EvSpIOE             processing complete
                  EvSpERROR           bad card
                  EvSpTIMOUT          no card


 SwCARD           EvSpIOE             processing complete
                  EvSpERROR           bad card
                  EvSpTIMOUT          no card
                  EvSpNEW             new card or uninitialized

---------------------------------------------------------------
```
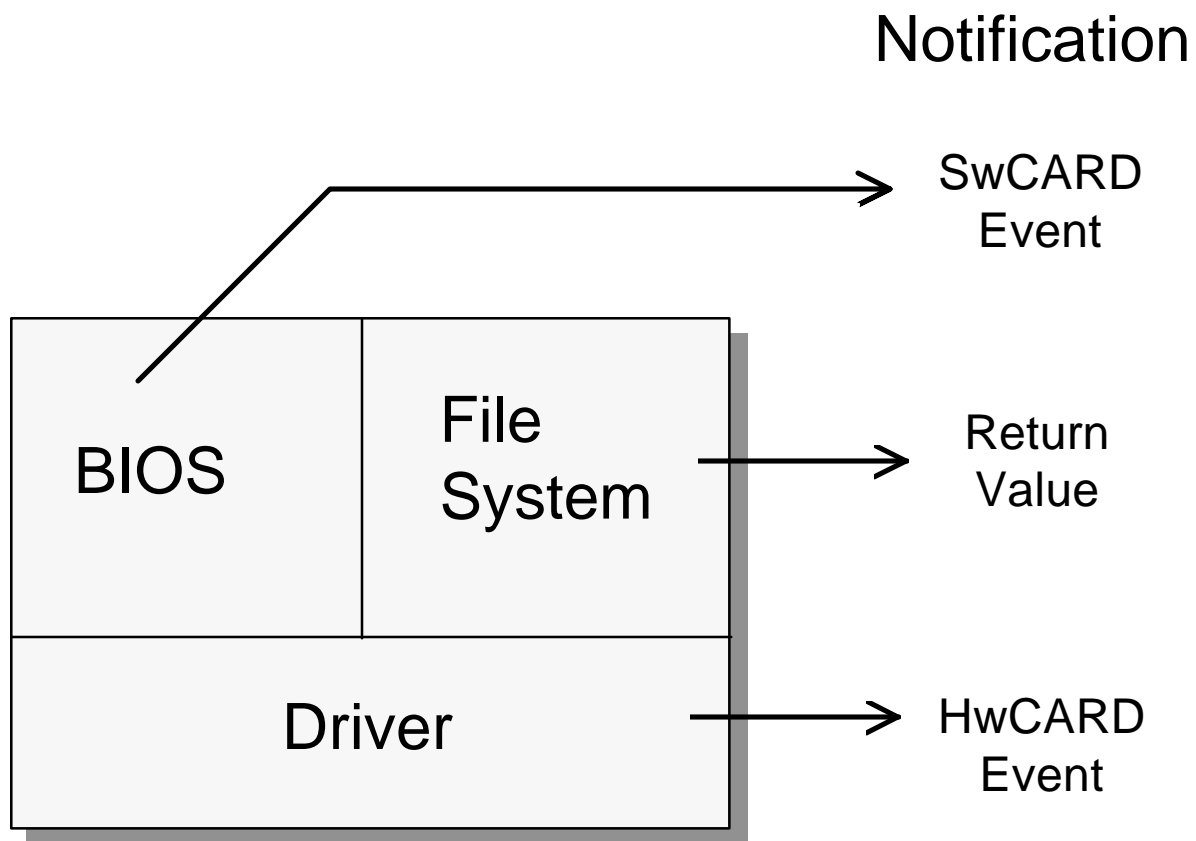
Error notification

Notification

SwCARD
Event

BIOS | File System → Return Value

Driver → HwCARD Event

File systems and error notification

```
================================================================
Function name          Only with directory cache errors   Always
----------------------------------------------------------------
  open                 HwCARD                             ----
  close                ----                               ----
  read/write           ----                               HwCARD

  format               ----                               ----
  firstfile            ----                               ----
  nextfile             ----                               ----
  delete               ----                               ----
  rename               ----                               ----

  _card_read           ----                               HwCARD
  _card_write          ----                               HwCARD
----------------------------------------------------------------
```

Real-time accessing

Device bu supports non-blocking mode.
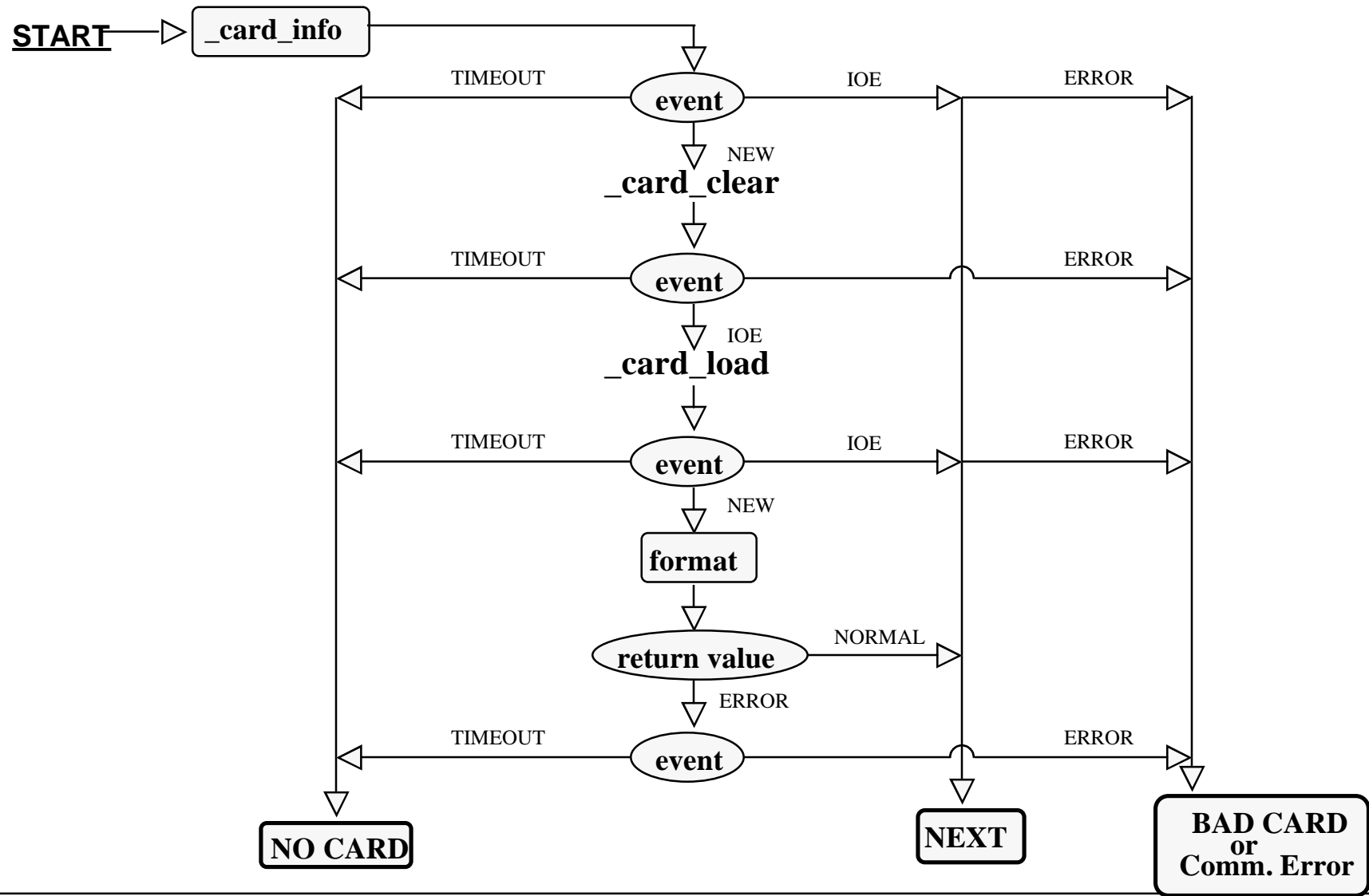➤ Specify 0_NOWAIT when open.

read()/write() sends an input/output request to the driver and then immediately exits.

Completion of input/output is notified as an event.

Each slot handles only one input/output request at a time.

Memory card BIOS / testing for card presence and formatting

START ▷ **_card_info**

TIMEOUT ◁————— **event** —————IOE—————————ERROR ▷

NEW ▽

_card_clear

TIMEOUT ◁————— **event** —————————————ERROR ▷

IOE ▽

_card_load

TIMEOUT ◁————— **event** —————IOE—————ERROR ▷

NEW ▽

format

return value —————NORMAL ▷

ERROR ▽

TIMEOUT ◁————— **event** —————————————ERROR ▷

NO CARD   **NEXT**   **BAD CARD or Comm. Error**

Filename

```
=========================================================
Byte              Contents           Notes
---------------------------------------------------------
0                 Magic              always 'B'
1                 Location           'I' is domestic
                                     'A' is North America
                                     'E' is Europe (*1)
2-11              Title              SCE part number (*2)
12-20             Available to       Any ASCII code except 0x00.
                  the user           End with 0x00.
---------------------------------------------------------
```

*1: No system checks on any of these

*2: For the first disk in multiple disk titles. For example, if the product code is "SLPS-0001", then the first 12 characters of the filename will be "BISLPS-00001". (The numerical portion is always expanded to 5 characters with zeros)

File header (1)

These should be located at the start of the data area

```
================================================
 Item                              Size (byte)
-------------------------------------------------

 Header                            128
     Magic                         2 (always 'SC')
     Type                          1
     Number of slots               1
     Name of document              64 (shift-JIS, *1)
     pad (padded with blank spaces) 28
     CLUT                          32
................................................

 Icon image (1)    128 (16 x 16 x 4 bits)
 Icon image (2)    128 (Type==0x12,0x13 only)
 Icon image (3)    128 (Type==0x13 only)

-------------------------------------------------
 *1: Non-kanji and Level 1 kanjis only. 32 full-width characters.
```

File header (2)

```
============================
Type    Number of icon images
        (automatic replacement
        animation)
----------------------------------------------
0x11    1
0x12    2
0x13    3
----------------------------------------------
```

Design issues

- Confirm card initialization with the operator

- Allow execution even without a card

Controller and memory card

Serial controller
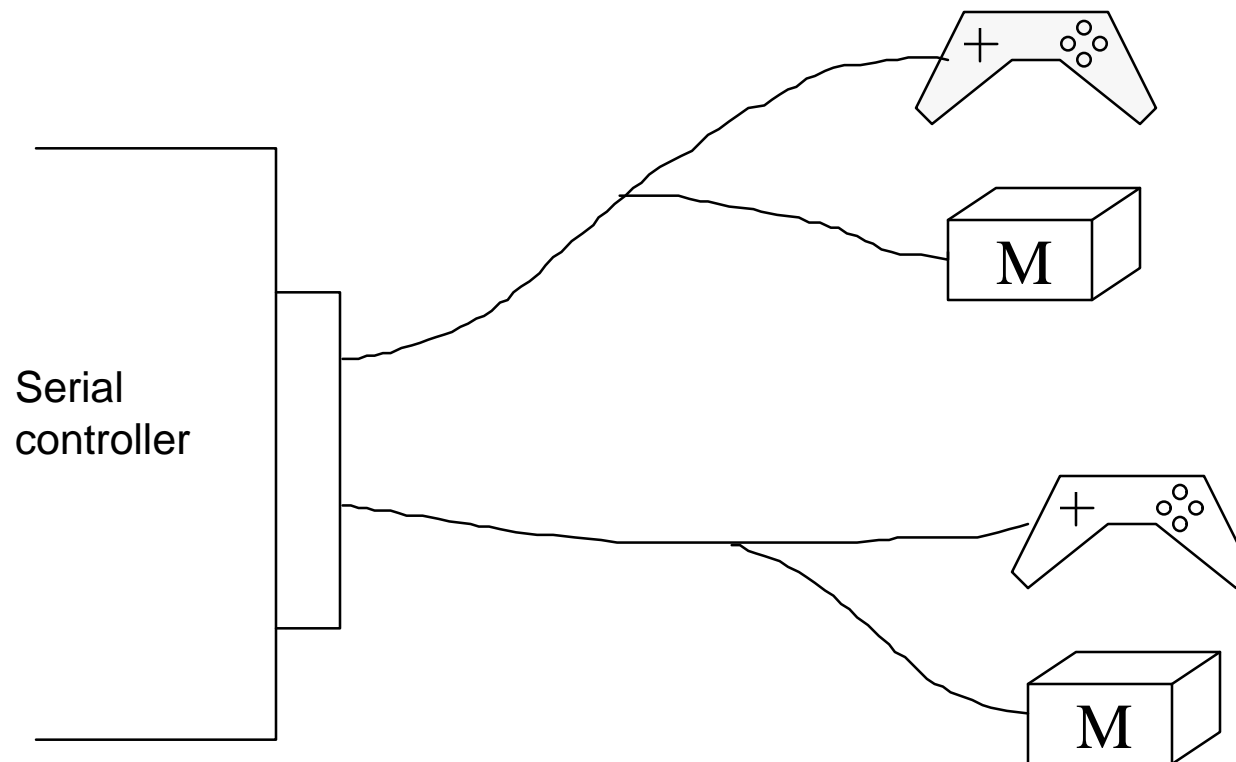
- Communication failure with memory card ➤ Communication failure with controller

Event drivers <reference>

OpenEvent(SwCARD or HwCARD)

BIOS $\longrightarrow$

event handler functions

$\longrightarrow$ DeliverEvent(DescUEV | n , spec)
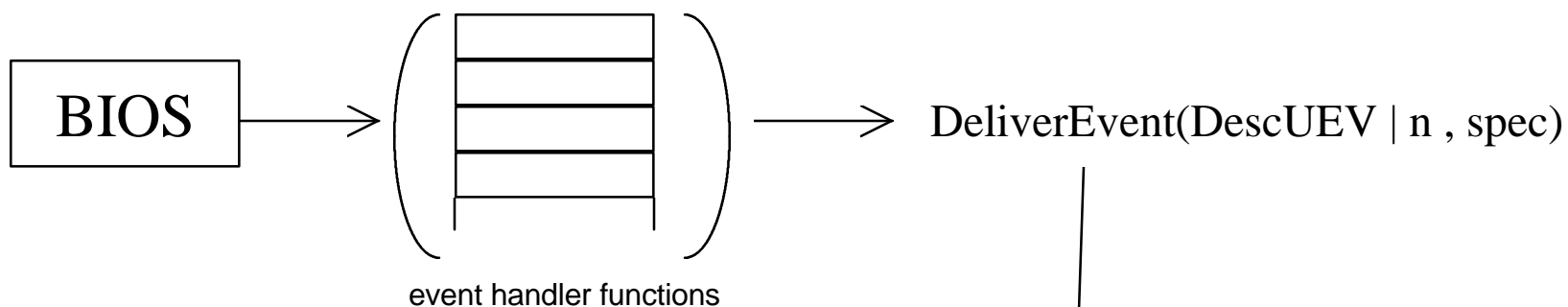
OpenEvent(DescUEV | n , spec , ...)

main flow   flag test/event handler functions

CONFIDENTIAL
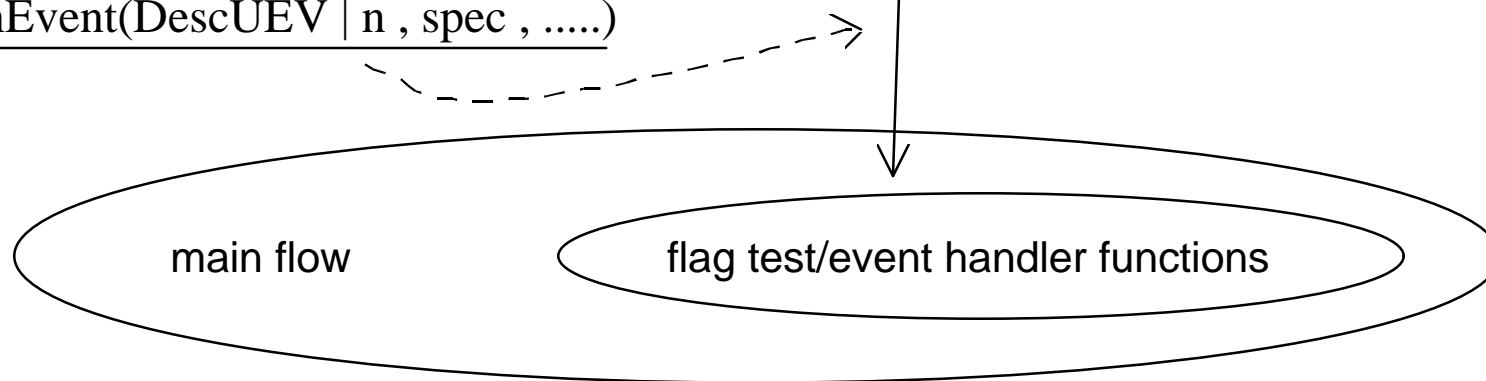
File:main.c

      main()   main function

      static void init_prim()   prepare for sprite display

      int pad_read()   controller access function

      dir_file()   get file directory

File:c_server.c

      CsHotStart()   hot start

      CsStop()   stop server

      CsInit()   cold start

      static _clear_event()   memory card function event clear

      static _ioe_handler()   SwCARD/EvSP10E handler

      static _new_handler()   SwCARD/EvSpNEW handler

      static _timeout_handler()   SwCARD/EvSpTIMOUT handler

      static _error_handler()   SwCARD/EvSpERROR handler

      static ReEntry()   Reactivate server (automatic reactivation entry)

      static _work1()   Reactivate server (main entry)

      Memory card controller API functions

      _init_pad()   initialization

      _get_cont()   repetitive call entry

      _copy_back()   restore input buffer