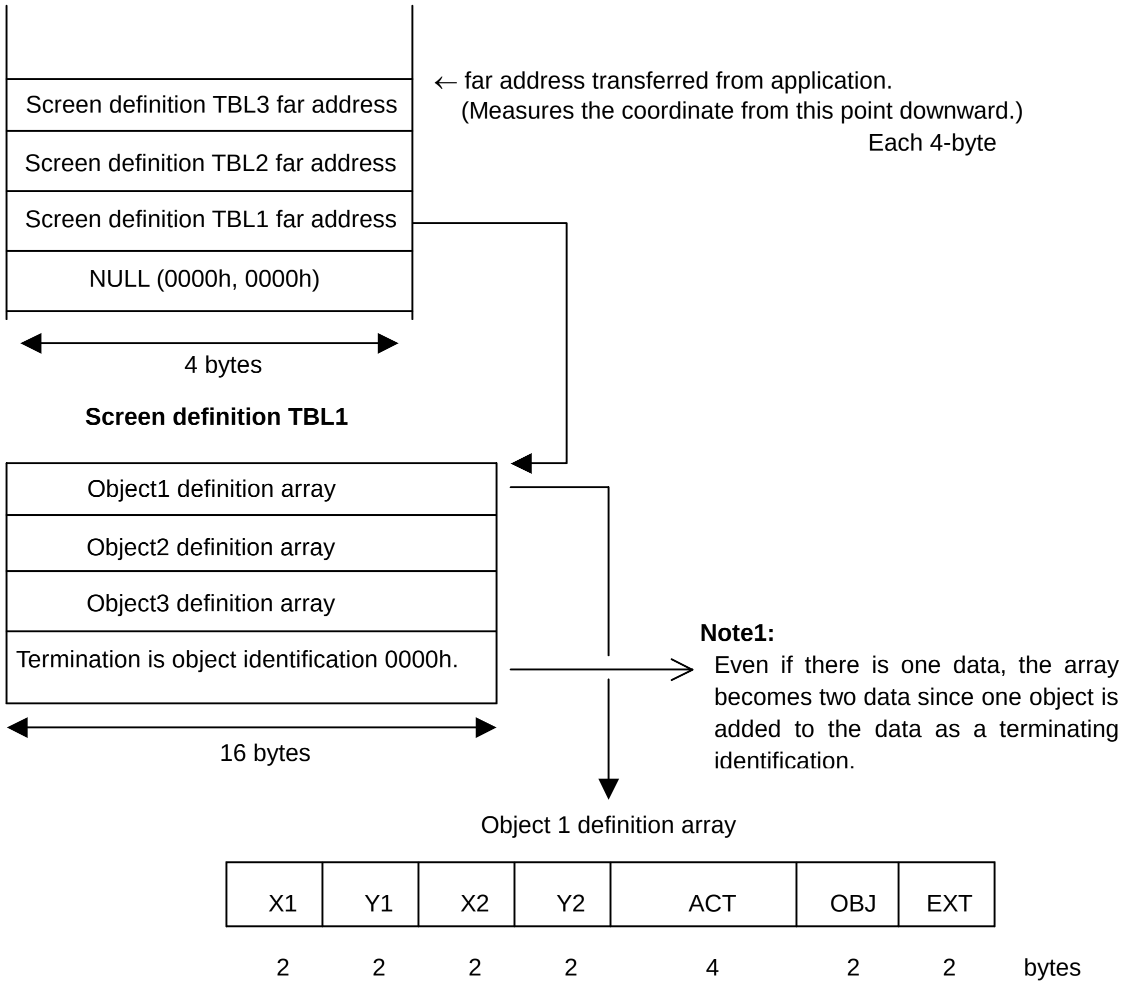
June 7, 1999

PocketViewer Event Control BIOS

[Outline of event control by screen definition]


Screen definition TBL3 far address
Screen definition TBL2 far address
Screen definition TBL1 far address
NULL (0000h, 0000h)
← far address transferred from application.
(Measures the coordinate from this point downward.)
Each 4-byte
4 bytes
Screen definition TBL1
Object1 definition array
Object2 definition array
Object3 definition array
Termination is object identification 0000h.
16 bytes
Object 1 definition array
X1
Y1
X2
Y2
ACT
OBJ
EXT
2
2
2
2
4
2
2
bytes

Note1:
Even if there is one data, the array becomes two data since one object is added to the data as a terminating identification.

Fig. 1: Configuration of object definition TBL

- Memory allocation of object definition data (array)

X1 0123h	Y1 4567h	X2 89ABh	Y2 CDEFh	ACT 12345678h	OBJ 9ABCh	EXT DEF0h
↓	↓	↓	↓	↓	↓	↓
23, 01,	67, 45,	AB, 89,	EF, CD,	78, 56, 34, 12,	BC, 9A,	F0, DE

Small ⟵ Address ⟶ Large

<Contents of object definition parameters>

- X1,Y1,X2,Y2 : Coordinates of object area (Square designation)
  Upper left coordinates (X1, Y1) - Lower right coordinates (X2, Y2)

- ACT Action identification code

(Code)	(Name)	(Contents of action)
00000001h	touch MaKe	; Moment when touching is made.
00000002h	touch MoVe	; Moved during touching
00000004h	touch MoVe OUT	; Moment when moved outside an object during touching
00000008h	touch MoVe IN	; Moment when moved inside an object during touching
00000010h	touch DoWn	; During touching
00000020h	touch DoWn IN	; Inside an object during touching
00000040h	touch BreaK	; Touching is released.
00000080h	touch BreaK IN	; Touching is released inside an object.
00000100h	touch REPeat	; Repeat interval time has elapsed during touching.
00000200h	500Msec	; 500MSEC update
00000400h	ALarM	; User alarm
00000000h	NONE space	; Invalid area is specified.

- OBJ: Object code

0000h	; NULL code (2-byte) is put to OBJ as a terminator (stopper) of screen definition TBL.
0001h - 7FFFh	; Reserved by system/Common
8000h - FFFFh	; Free setting by application

- EXT Extension code

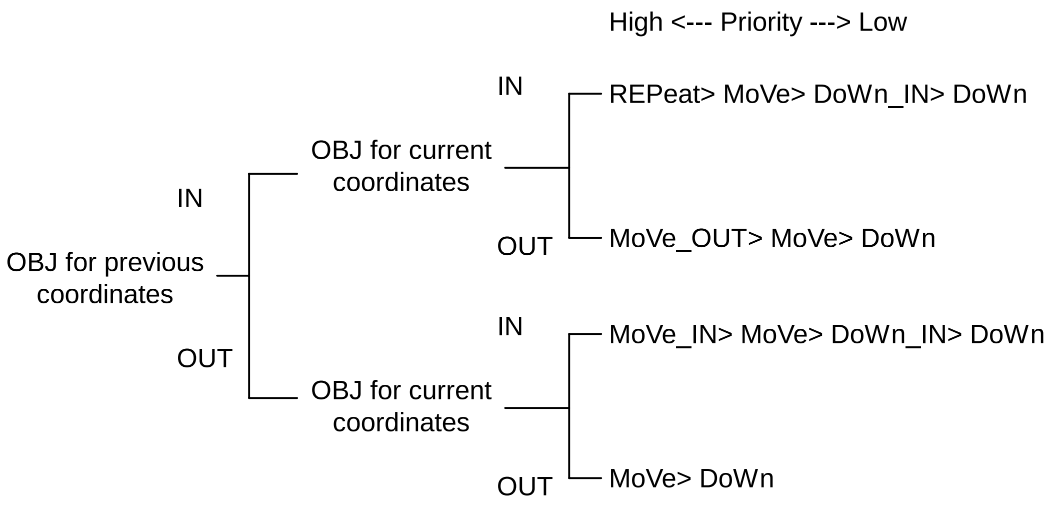
0000h	; Reserved code for extension (Normally, this code is NULL.)

<Outputting an occurrence object>

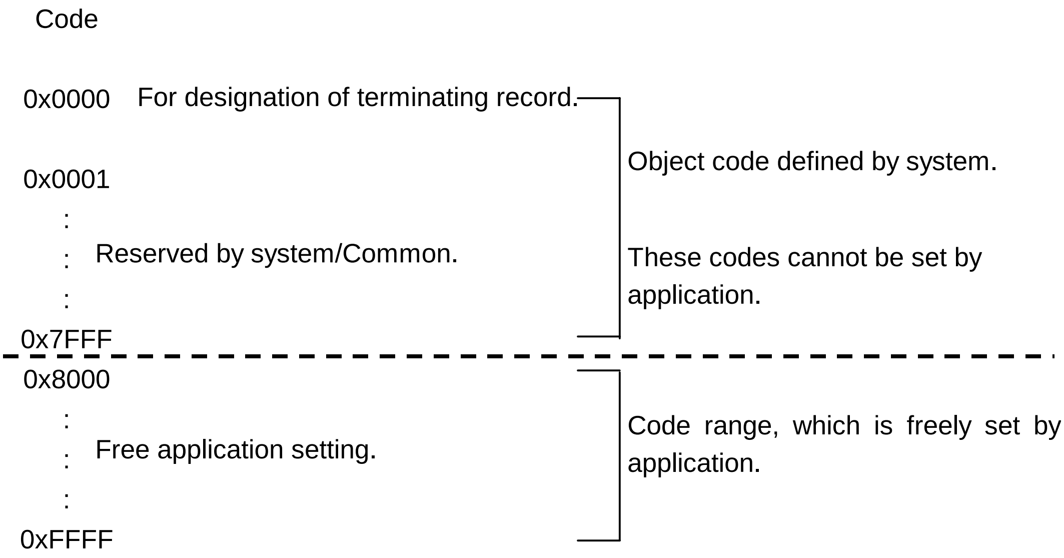
- Contents of pointer for output structure

(Member)				
ROBJ	(DW)	Occurrence object	2-byte	Object in which event occurs.
RACTL	(DW)	Occurrence action (LOW)	2-byte	Any action of object occurred (Only 1 bit is ON.)
RACTH	(DW)	Occurrence action (HIGH)	2-byte	(RACTL shows the LOW side and RACTH shows the HIGH side.)
RX	(DW)	Touch X-coordinate	2-byte	X-coordinate of current touch position
RY	(DW)	Touch Y-coordinate	2-byte	Y-coordinate of current touch position
REXT	(DW)	Additional occurrence information	2-byte	Normally, this is NULL and used by 500mSEC.
RISTR	(DB)	For expansion (Normally, NULL)	4-byte	This is kept for expansion and normally NULL.

- Outline of action identification priority
  (After MAKE, while coordinates are obtained by continuous input (INK).)


High <--- Priority ---> Low
OBJ for previous coordinates
IN
OUT
OBJ for current coordinates
IN
REPeat> MoVe> DoWn_IN> DoWn
OUT
MoVe_OUT> MoVe> DoWn
OBJ for current coordinates
IN
MoVe_IN> MoVe> DoWn_IN> DoWn
OUT
MoVe> DoWn

[Classification of object identification codes]



Code

0x0000 For designation of terminating record.

0x0001
:
: Reserved by system/Common.
:
0x7FFF

Object code defined by system.

These codes cannot be set by application.

0x8000
:
: Free application setting.
:
0xFFFF

Code range, which is freely set by application.

<<End record designation (0x0000) >>

This object is to be set at the end of the object definition array as the terminator of the screen definition TBL in order to determine the end position.

[Object code]	[Name]	[EQU]	Remarks
0x0000	TBL end	IW_OBJ_END	Termination code of screen definition TBL

<<System reserved/Common items (0x0001 - 0x7FFF)>>

Objects, which depend on the hardware, such as hard icons, and those, which are used in BIOS to transfer them from the system to applications, are classified and defined into the following groups.

- Objects in hard icon (OFF or menu, etc.)
- Lever-push switch objects
- 500mSEC output objects
- Alarm match objects
- Special objects, such as break key sampling or touch scanning

♦**<Hard icons>** See also <<Appendix 1>>.

As the name expresses, these objects are those already printed on the panel.

Object code: 0x0001 - 0x0009
Object code: 0x0021 - 0x0029 (When powered ON)

;** Object code of hard icon **

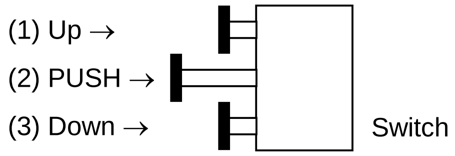
<table>
<tr><td>(2)</td><td rowspan="2">(1)</td><td rowspan="2">(4)</td><td rowspan="2">(5)</td><td rowspan="2">(6)</td><td rowspan="2">(7)</td><td>(8)</td></tr>
<tr><td>(3)</td><td>(9)</td></tr>
</table>

[Object code]	[Name]	[EQU]		Remarks
0x0001	Hard-icon 1	IW_HARDICON1	(1)	MENU
0x0002	Hard-icon 2	IW_HARDICON2	(2)	OFF
0x0003	Hard-icon 3	IW_HARDICON3	(3)	EL
0x0004	Hard-icon 4	IW_HARDICON4	(4)	SCHEDULER
0x0005	Hard-icon 5	IW_HARDICON5	(5)	CONTACTS
0x0006	Hard-icon 6	IW_HARDICON6	(6)	MEMO
0x0007	Hard-icon 7	IW_HARDICON7	(7)	MBAR
0x0008	Hard-icon 8	IW_HARDICON8	(8)	ESC
0x0009	Hard-icon 9	IW_HARDICON9	(9)	QUICKMEMO

(Hard-icon objects when power is turned on.)

[Object code]	[Name]	[EQU]		Remarks
0x0021	Hard-icon 1	IW_HDICON1_PON	(1)	MENU
(0x0022	Hard-icon 2	IW_HDICON2_PON	(2)	OFF)
(0x0023	Hard-icon 3	IW_HDICON3_PON	(3)	EL)
0x0024	Hard-icon 4	IW_HDICON4_PON	(4)	SCHEDULER
0x0025	Hard-icon 5	IW_HDICON5_PON	(5)	CONTACTS
0x0026	Hard-icon 6	IW_HDICON6_PON	(6)	MEMO
0x0027	Hard-icon 7	IW_HDICON7_PON	(7)	MBAR
(0x0028	Hard-icon 8	IW_HDICON8_PON	(8)	ESC)
0x0029	Hard-icon 9	IW_HDICON9_PON	(9)	QUICKMEMO
0x0030	Cradle KeyPON	IW_CRDLKEY_PON		Unit is powered ON by pressing CRADLE key.

♦ <**Lever-Push Switch>** See also <<Appendix 2>>.



[Object code]	[Name]	[EQU]	Remarks
0x0011	Up	IW_LPSW_UP	(1) Up side
0x0012	PUSH	IW_LPSW_PUSH	(2) PUSH
0x0013	Down	IW_LPSW_DOWN	(3) Down side
0x0015	PUSH ON	IW_LPSW_PON	Power ON by Lever-Push Switch

♦ <**500mSEC output** (Individual occurrence)> See also <<Appendix 3>>.

[Object code]	[Name]	[EQU]	Remarks
0x001F	500Msec event	IW_500MSEC	Outputs depending on the clock count of the CPU.

♦ <**Alarm output** (individual occurrence)> See also <<Appendix 4>>.

[Object code]	[Name]	[EQU]	Remarks
0x5007	Alarm match event	IW_SYS_ALARM	Matching with alarm set as next alarm

♦ <**Break key sampling or touch scanning>** See also <<Appendix 5>>.

[Object code]	[Name]	[EQU]	Remarks
0x5000	No event	IW_NOEVENT	No event occurrence (Touch scanning/Break key sampling)
0x5001	BLD message level	IW_SYS_BLD1	BLD1 (BLD message level)
0x5003	Press Cradle key	IW_SYS_CRDLKEY	Cradle key is pressed.
0x5009	ESC	IW_SYS_ESCTCH	ESC is touched during brake key sampling

<<Appendix 1>> (Hard icons)

- Coordinates of hard icons included in the main unit are shown below.

<table>
<tr><td>(2)</td><td rowspan="2">(1)</td><td rowspan="2">(4)</td><td rowspan="2">(5)</td><td rowspan="2">(6)</td><td rowspan="2">(7)</td><td>(8)</td></tr>
<tr><td>(3)</td><td>(9)</td></tr>
</table>

<Common versions for simulator and actual unit>

	Start coordinates (upper left) (X, Y)	End coordinates (Lower right) (X, Y)	Object code	
(1) MENU	(17,164)	(41,200)	IW_HARDICON1	(0x0001)
(2) OFF	(-8,164)	(16,177)	IW_HARDICON2	(0x0002)
(3) EL (OFF)	(-8,178)	(16,200)	IW_HARDICON3	(0x0003)
(4) SCHEDULER	(42,164)	(65,200)	IW_HARDICON4	(0x0004)
(5) CONTACTS	(66,164)	(90,200)	IW_HARDICON5	(0x0005)
(6) MEMO	(91,164)	(114,200)	IW_HARDICON6	(0x0006)
(7) MBAR	(140,178)	(164,200)	IW_HARDICON7	(0x0007)
(8) ESC	(140,164)	(164,177)	IW_HARDICON8	(0x0008)
(9) QUICKMEMO	(115,164)	(139,200)	IW_HARDICON9	(0x0009)

<<Appendix 2>> (Lever-Push Switch)

<Input>

- Action control can be used by defining an object in the same manner as described for hard icons or other events. The object definitions are shown in the TABLE below.

	X1	Y1	X2	Y2	ACT	OBJ	EXT			
	2	2	2	2	4	2	2	← bytes		
DW	00000h,	00000h,	00000h,	00000h,	00101h, 00000h,	00011h,	00000h	Up	(1)	Up side
DW	00000h,	00000h,	00000h,	00000h,	00001h, 00000h,	00012h,	00000h	PUSH	(2)	PUSH
DW	00000h,	00000h,	00000h,	00000h,	00101h, 00000h,	00013h,	00000h	Down	(3)	Down side
	<--->				<--------------->	<----->	<-------->			
	Coordinate value (NULL)				Action	Object	Addional infomation			

<Output>

- The following shows the contents of structure output when the lever-push switch (action control) is pressed.

ES:[DI].ROBJ	→	0x0011 - 0x0013 or 0x0015
ES:[DI].RACTL	→	IX_ACT_MKL(0x0001) or IX_ACT_REPL(0x0100)
ES:[DI].RACTH	→	IX_ACT_MKH(0x0000) or IX_ACT_REPH(0x0000)
ES:[DI].RX	→	; Touch X-coordinate (Invalid)
ES:[DI].RY	→	; Touch Y-coordinate (Invalid)
ES:[DI].REXT	→	; Additional information
ES:[DI].RISTR	→	; NULL (4-byte)

<Remarks>

Since no coordinate values are needed during input, the coordinate values are fixed at NULL. Therefore, above three coordinates are defined as one set.
Additionally, since (2) PUSH is not repeated, action designation is made only on MAKE. It is not necessary to define power ON by lever-push in the touch stack area. If action occurs, the object 0x0015 is output unconditionally.

<<Appendix 3>> (500mSEC event)

<Input> Object definition of screen definition table

	X1	Y1	X2	Y2	ACT	OBJ	EXT	
	2	2	2	2	4	2	2	← bytes
DW	00000h,	00000h,	00000h,	00000h,	00200h, 00000h,	0001Fh,	00000h	; 500 msec

```
    <---------------------------------------->   <--------------->  <------->
              Coordinate value (NULL)                 Action        Object
                                                   (0x00000200)    (IW_500MSEC = 0x001F)
                                                                                <--------->
                                                                                Addional infomation (NULL)
```

<Output>

(1) 500mSEC event occurs individually.

ES:[DI].ROBJ	→	IW_500MSEC	(0x001F)
ES:[DI].RACTL	→	IX_ACT_500ML	(0x0200)
ES:[DI].RACTH	→	IX_ACT_500MH	(0X0000)
ES:[DI].RX	→	; Touch X-coordinate	Unknown (Invalid)
ES:[DI].RY	→	; Touch Y-coordinate	Unknown (Invalid)
ES:[DI].REXT	→	IW_EXT500M	(0x0001)
ES:[DI].RISTR	→	;NULL	(4-byte)

(2) 500mSEC event occurs together with other object event at the same time.

ES:[DI].ROBJ	→	Relevant object code	
ES:[DI].RACTL	→	Action of relevant object code (No valid actions → IX_ACT_500ML)	
ES:[DI].RACTH	→	Action of relevant object code (No valid actions → IX_ACT_500MH)	
ES:[DI].RX	→	Touch X-coordinate of relevant object	
ES:[DI].RY	→	Touch Y-coordinate of relevant object	
ES:[DI].REXT	→	IW_EXT500M	(0x0001)
ES:[DI].RISTR	→	NULL	(4-byte)

<Remarks>

By adding object definitions described in above <Input> to the screen definition TABLE, which is used to call up normal event BIOS, the contents shown in <Output> are output since it is determined as if 500mSEC event output is specified by the application.

<Limitations>

Since the 500mSEC count depends on the hardware timer (CPU clock), outputs are made at 1/2 cycle of updating of seconds in the clock.

<<Appendix 4>> (Alarm match event)

<Input>
The precondition is that the next alarm process is made by the alarm setting BIOS. No other conditions are particularly required.
It is not necessary to set alarm actions (0x00000400) specified by the output action in the screen definition TABLE.

<Output>
The following shows event occurrence outputs if the alarm is matched in the event BIOS (key waiting).

(1) Alarm match event occurs individually (in case of waiting for touch MAKE).

ES:[DI].ROBJ	→	IW_SYS_ALARM	(0x5007)
ES:[DI].RACTL	→	IX_ACT_ALML	(0x0400)
ES:[DI].RACTH	→	IX_ACT_ALMH	(0X0000)
ES:[DI].RX	→	; Touch X-coordinate	Unknown (Invalid)
ES:[DI].RY	→	; Touch Y-coordinate	Unknown (Invalid)
ES:[DI].REXT	→	; Added information	Unknown (Invalid)
ES:[DI].RISTR	→	; For expansion	Unknown (Invalid)

(2) Alarm match event occurs when operated with object defined by application.

ES:[DI].ROBJ	→	Relevent object code	
ES:[DI].RACTL	→	IX_ACT_ALML	(0x0400)
ES:[DI].RACTH	→	IX_ACT_ALMH	(0X0000)
ES:[DI].RX	→	Touch X-coordinate	(0x8000)
ES:[DI].RY	→	Touch Y-coordinate	(0x8000)
ES:[DI].REXT	→	; Added information	Unknown (Invalid)
ES:[DI].RISTR	→	; For expansion	Unknown (Invalid)

<Remarks>
Output (1) occurs if the panel is not touched or if touching of the NOP area is released with the unit put in the waiting status. The object outputs IW_SYS_ALARM (0x5007) in the system definition and the action code outputs 0x00000400. Values of the structure for other outputs are invalid (unknown).

Output (2) is those when the alarm match occurs while an object defined in the application, such as action during touching (MoVe, DoWn, REPeat, etc.) is functioning. If this occurs, the currently relevant (currently operating) object is set and 0x8000 (invalid value) is set in the touch coordinate value.
At this time, it is necessary to make relevant object invalid in the application (such as cancellation of reverse display).

<<Appendix 5>> (Break key sampling)

- Examples of all-data registration (communication), all-data deletion, and search

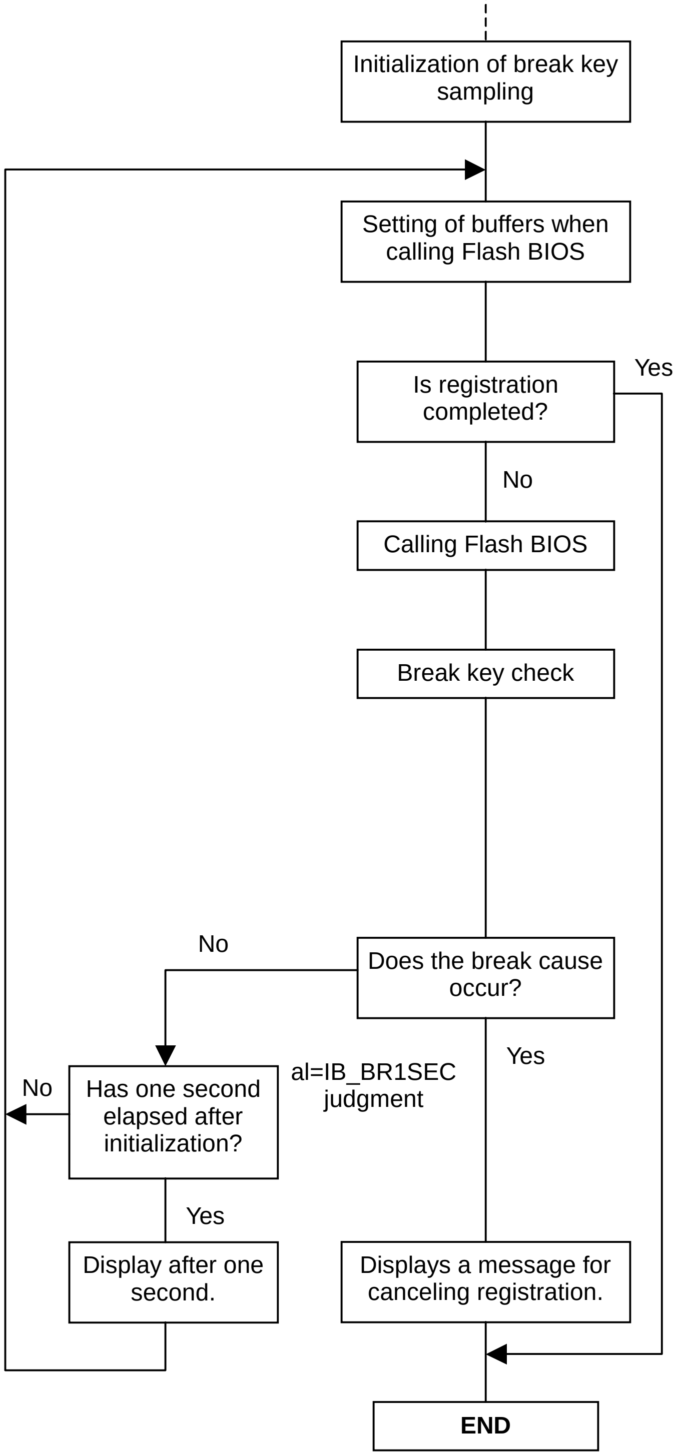
```
IB_BRSAMP          (0x06)     ;Break Key sampling (Accessing to FLASH BIOS)
IB_EVTLIB          (0x50)     ;Event control LIB

;*** BREAK KEY sampling ***
IX_BRSAMP_CK       (0x80)     ;Checking of cause of break (Other bits are invalid.)
IX_BRSAMP_INIT     (0x40)     ;Initialization of BREAK KEY sampling (bits 0 - 3 are valid.)
IX_BLD1MSG         (0x08)     ;BLD message level
IX_CRADLE          (0x04)     ;Cradle KEY
IX_ESCBRK          (0x01)     ;ESC key

;*** 1 sec. lapse flag after initialization ***
IB_NULL            (0x00)     ;1 sec. has not elapsed after initialization.
IB_BR1SEC          (0x01)     ;1 sec. has elapsed after initialization.

;*** Cause of break ***
IW_NOEVENT         (0x5000)   ;Cause of break does not occur.
IW_SYS_CRDLKEY     (0x5003)   ;Synchronization (cradle) KEY is pressed.
IW_SYS_BLD1        (0x5001)   ;BLD message level
IW_SYS_ESCTCH      (0x5009)   ;"ESC" touch
```

Character input process, etc.


Initialization of break key sampling
Setting of buffers when calling Flash BIOS
Is registration completed?
Yes
No
Calling Flash BIOS
Break key check
Does the break cause occur?
No
Yes
Has one second elapsed after initialization?
al=IB_BR1SEC judgment
No
Yes
Display after one second.
Displays a message for canceling registration.
END

ah <- IB_BRSAMP
al <- IX_BRSAMP_INIT
(Following is made by OR condition.)
IX_BLD1MSG IX_CRADLE
IX_ESCBRK
ds:si <- Coordinate value of ESC icon * Note1

Note 1: Start (X,Y) - End (X,Y) makes 2 icons, 8 words in total. In case of one icon, the same value is put to make 8 words in total.
Additionally, both X and Y coordinate ranges do not cross over 0 dot.

Example: Calling of flash BIOS for registration of one record.

ah <- IB_BRSAMP
al <- IX_BRSAMP_CK
INT EVTLIB

<OUT>
dx (Cause of break)
IW_NOEVENT ; No cause of break occurs.
IW_SYS_CRDLKEY; Synchronous KEY is pressed.
IW_SYS_BLD1; BLD message level
IW_SYS_ESCTCH ;"ESC" touch
al (1 sec. lapse flag)
IB_NULL ; Not elapsed.
IB_BR1SEC ; Elapsed
Judged by dx=IW_NOEVENT.

If 1 sec. message is already displayed, display is started after closing WINDOW.