Rule of Constant for OMR STATUS

With the exemption of SR_SUCCESS, the smaller the number displayed, the more serious the error. There are two types of status information: for the front sensor unit and for the back sensor unit. The higher priority item is selected, and if both front and back sensor unit items are at the same level of priority, the front sensor unit will be selected first.

Bit31	Priority (4 Bits)
:	0x0 : Hardware Error (release disabled)
:	0x1 : Connection Error
:	0x2 : Cover Open
:	0x3 : Paper Jam
Bit28	0x4 : Warning/Operation error
Bit27	Problematic Area (4 Bits)
:	0x0 : Main body
:	0x1 : Front Sensor Unit
:	0x2 : Back Sensor Unit
:	0x3 : Barcode Unit
:	0x4 : Printer Unit
:	0x5 : Stacker Unit
Bit24	0xf : Others
Bit23	Page Number (8 Bits)
:	The pages are divided in alphabetical order according to the first digit of the status infomation.
:	0x00 : Error during Communication (an error occurring prior to gaining status information)
:	0x01 : Status information of 1st digit=A
:	:
Bit16	0x1A : Status information of 1st digit=Z
Bit15	Through Number (16 Bits)
:	Through number per each page
Bit0	

4.2 API System Control

4.2.1 OMR_OpenDeviceUSB

Prototype	OMR_STATUS OMR_OpenDeviceUSB(void)	
Process	Detects a device connected to the USB and opens the device.	
Parameter	None	
Response Value	SR_SUCCESS	Successful
	SR_UNSUCCESSFUL	Failure (there is no device that can be opened, or is preoccupied by another connection).
Details	When a multiple number of OMR devices are connected, internal control will allow priority connection with the initial OMR device.	

4.2.2 OMR_CloseDevice

Prototype	OMR_STATUS OMR_CloseDevice(void)	
Process	Closes a device handler opened by an OMR_OpenDeviceUSB function. Must be conducted when closing down an application.	
Parameter	None	
Response Value	SR_SUCCESS	Successful
	SR_UNSUCCESSFUL	Failure

4.2.3 OMR_GetLastError

Prototype	OMR_STATUS OMR_GetLastError(void)
Process	Most control API can only retrieve success or failure response values. If unsucoessful, one method to find the cause is to use the OMR_STATUS function value as the last recorded data.
Parameter	None
Response Value	The last recorded OMR_STATUS value
Details	The OMR_STATUS is defined as typedef unsigned int OMR_STATUS. Please refer to the "Constant" section for further details on storage. When executing OMR_OpenDeviceUSB/OMR_CloseDevice/OMR_GetLastError, the OMR_STATUS will not be recorded.

4.2.4 OMR_FormatMessage

Prototype	CHAR *OMR_FormatMessage (OMR_STATUS status, int iLanguageFlag)	
Prosess	Convert OMR_STATUS value into text string.	
Parameter	status	OMR_STATUS value to be converted
	iLanguageFlag	Output Language Setting SR_STRING_NORMAL: not defined (English) SR_STRING_ENGLISH: English (only ASCII Code) SR_STRING_JAPANESE: Japanese (Shift-JIS Code)
Return Value	Pointer to the converted text string (fixed value)	
Details	Refer to the OMR_STATUS constant list for conversion results	
Example	The following usage is possible combined with OMR_GetLastEerror. print(OMR_Format Message (OMR_GetLastError0, ST_STRING_NORMAL	

4.2.5 OMR_GetST

Prototype	const CHAR *OMR_GetST(int iPage)	
Prosess	Directly output status information received during last response.	
Parameter	iPage	SR_PAGE_FRONT : Assign ST1 data SR_PAGE_BACK : Assign ST2 data
Return Value	Pointer to status data text string (fixed value). Double bite text string. If the text string is empty (text lwnfth is 0), there is no data or the deduction value is incorrect.	
Details	Example PRINT("%s"),OMR_GetST(SR_PAGE_FRONT));	