

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 information.
:	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 unsuccessful, 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)	
Process	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_GetLastError. print(OMR_Format Message (OMR_GetLastError0, ST_STRING_NORMAL	

4.2.5 OMR_GetST

Prototype	const CHAR *OMR_GetST(int iPage)	
Process	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 length is 0), there is no data or the deduction value is incorrect.	
Details	Example PRINT("%s"),OMR_GetST(SR_PAGE_FRONT));	

4.3 Read Settings

4.3.1 OMR_SetNumberOfColumnsToRead

Prototype	BOOL OMR_SetNumberOfColumnsToRead(int iColumns)																						
Process	Use SetNumberOfColumnsToRead comand to use the value set by the parameter to set the number of columns to read.																						
Parameter	iColumns	Set the designated line number. If selecting setting the value of 0, it will return to the initial setting.																					
Return Value	TRUE	Successful																					
	FALSE	Unsuccessful																					
Details	The line setting will differ depending on the reader unit instaled in each OMR hardware. <table><thead><tr><th>Reader Unit Sensor Pitch</th><th>Setting Value</th><th>Initial Value</th></tr></thead><tbody><tr><td>1/6 inch</td><td>1 - 48</td><td>48</td></tr><tr><td>0.2 inch</td><td>1 - 40</td><td>40</td></tr><tr><td>0.2 inch S</td><td>1 - 40</td><td>40</td></tr><tr><td>0.25 inch</td><td>1 - 33</td><td>33</td></tr><tr><td>0.3 inch</td><td>1 - 27</td><td>27</td></tr><tr><td>0.3 inch F</td><td>1 - 24</td><td>24</td></tr></tbody></table>		Reader Unit Sensor Pitch	Setting Value	Initial Value	1/6 inch	1 - 48	48	0.2 inch	1 - 40	40	0.2 inch S	1 - 40	40	0.25 inch	1 - 33	33	0.3 inch	1 - 27	27	0.3 inch F	1 - 24	24
Reader Unit Sensor Pitch	Setting Value	Initial Value																					
1/6 inch	1 - 48	48																					
0.2 inch	1 - 40	40																					
0.2 inch S	1 - 40	40																					
0.25 inch	1 - 33	33																					
0.3 inch	1 - 27	27																					
0.3 inch F	1 - 24	24																					

4.3.2 OMR_GetNumberOfColumnsToRead

Prototype	int OMR_GetNumberOfColumnsToRead(void)	
Process	This command is used to get the number of columns to be read that have been set by the OMR	
Parameter	None	
Return Value	0	Failure
	Other than 0	Number of columns as set by the OMR

4.3.3 OMR_SetReadingMethod

Prototype	BOOL OMR_SetReadingMethod(int iControlType, int iMultipleValue)	
Process	This command RM sets the method which the OMR reads marks.	
Parameter	iControlType	<p>Set read control values</p> <p>SR_READ_INITIAL : Return to initial value (direct control value=3)</p> <p>SR_READ_FRONT_EDGE: Front edge control method</p> <p>SR_READ_REAR_EDGE: Rear edge control method</p> <p>SR_READ_DIRECT : Direct method</p> <p>SR_READ_FACOM : FACOM method</p> <p>SR_READ_BETWEEN_MARK_NO_SPACE : No mark space method (without reading front edge margin)</p> <p>SR_READ_BETWEEN_MARK : Between mark method (with reading of front margins)</p>
	iMultiple Value	<p>Multiple Control Front control type is set from 1-9 Rear control type is set from 2-9</p> <p>However, these values are ignored when not being a control type.</p>
Return Value	TRUE	Successful
	FALSE	Failure

4.3.4 OMR_GetReadingMethod

Prototype	BOOL OMR_GetReadingMethod(int *iControlType, int *iMultipleValue)	
Process	The set value can be gained by selecting the reading method command RM.	
Parameter	*iControlType	An address that stores the value of the reading method command RM.
	*iMultipleValue	The address that stores the control multiple value that was read.
Return Value	TRUE	Successful
	FALSE	Failure
Details	<p>Example</p> <pre>int ctl_type, multi_val; //Gain and execute OMR_GetReadingMethod(&ctl_type, &multi_val)</pre>	

4.3.5 OMR_SetBackSensorUnit

Prototype	BOOL OMR_SetBackSensorUnit(int iDirective)	
Process	The Set Back Sensor Unit command can be used to set whether or not the Back Sensor Unit should be used.	
Parameter	iDirective	Command setting SR_INTERNAL : reset to initial setting (for use) SR_ENABLE : Setting enabled SR_DISABLE : Setting disabled
Return Value	TRUE	Successful
	FALSE	Failure

4.3.6 OMR_GetBackSensorUnit

Prototype	int OMR_GetBackSensorUnit(void)	
Process	Use the Get Back Sensor Unit command to attain the enabling, disabling of the back reading sensor unit	
Parameter	None	
Return Value	SR_ENABLE	Setting enabled
	SR_DISABLE	Setting disabled

4.3.9 OMR SetSheetThickness

Prototype	BOOL OMR_SetSheetThickness(int iThickness)	
Process	Set the ream weight (thickness) using the ream weight setting command FT.	
Parameter	iThickness	Sheet ream weight (thickness) setting SR_THICKNESS_AUTO_DETECT : Automatic detection SR_THICKNESS_64_GPM2 : 64g/m2(55 kg sheet) SR_THICKNESS_84_GPM2 : 84g/m2(72 kg sheet) SR_THICKNESS_105_GPM2 : 105g/m2(90 kg sheet) SR_THICKNESS_128_GPM2 : 128g/m2(110 kg sheet) SR_THICKNESS_157_GPM2 : 157g/m2(135 kg sheet) SR_THICKNESS_INTERNAL : Return to default value (90 kg sheet)
Return Value	TRUE	Successful
	FALSE	Failure

4.3.10 OMR_GetSheetThickness

Prototype	int OMR_SetSheetThickness(void)	
Process	Get the ream weight (thickness) using the ream weightsetting command FT.	
Parameter	None	
Return Value	SR_FUNCTION_FAIL	Error
	Other than mentioned above	Sheet ream weight (thickness)
		SR_THICKNESS_AUTO_DETECT : Automatic detection
		SR_THICKNESS_64_GPM2 : 64g/m2(55 kg sheet)
		SR_THICKNESS_84_GPM2 : 84g/m2(72 kg sheet)
		SR_THICKNESS_105_GPM2 : 105g/m2(90 kg sheet)
		SR_THICKNESS_128_GPM2 : 128g/m2(110 kg sheet)
		SR_THICKNESS_157_GPM2 : 157g/m2(135 kg sheet)

4.3.7 OMR_SetSheetFeedMode

Prototype	BOOL OMR_SetSheetFeedMode(int iMode, int iInsertTime)	
Process	Set paper feed mode by using paper feed mode command FM.	
Parameter	iMode	Paper feed mode setting. The following settings are required. SR_MODE_AUTO : Automatic paper feed mode SR_MODE_MANUAL : Manual paper feed mode SR_INITIAL : Return to initial setting (automatic paper feed mode timing 10[sec])
	iInsertTime	Sheet insert time. 0-99[sec]. When set to 0, there is no time limit. Other than manual feed mode, settings are denied.
Return Value	TRUE	Successful
	FALSE	Failure

4.3.8 OMR_GetSheetFreeMode

Prototype	BOOL OMR_GetSheetFeedMode(int *iMode, int *iInsertTime)	
Process	Set feed mode by using feed mode command FM.	
Parameter	*iMode	Storage address for of set feed mode. SR_MODE_AUTO : Automatic paper feed mode SR_MODE_MANUAL : Manual paper feed mode
	*iInsertTime	Address for set sheet insert timing
Return Value	TRUE	Successful
	FALSE	Failure
Details	<pre> Example int feed_mode, insert_time; //Execute OMR_GetsheetFeedMode(&feed_mode,&insert_time); </pre>	