

The terminal software for the Windows

AJATERM — MACRO

INTERNAL FUNCTION REFERENCE

(Ver 1.6.3.0)

— CONTENT —

1.	Option.....	1
2.	Screen information.....	1
3.	Display.....	2
4.	Array.....	3
5.	Integer operation.....	3
6.	Arithmetic operation.....	3
7.	Timer/Wait.....	4
8.	Timer Start / Stop.....	5
9.	String.....	5
10.	Accept.....	7
11.	Memory block.....	7
12.	File / Folder.....	8
13.	Acquire file name / folder name.....	9
14.	Profile.....	10
15.	Com-port / Mail-slot.....	11
16.	COM-port / Mail-slot event processing function.....	13
17.	File transfer by XMODEM / YMODEM.....	14
18.	Standard event processing function.....	14
19.	AJATERM connection event processing function.....	15
20.	Enable / Disable Event.....	16
21.	Button.....	17
22.	Check-Box.....	17
23.	Timechart graph.....	18
24.	Others.....	19
25.	Debug.....	20
26.	Sample macro text list.....	21

Attention

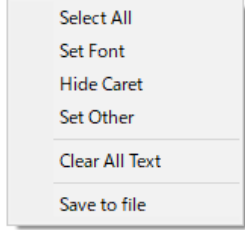
This manual is a Englishversion manual created by general Japanese to English translation software.
I think there are parts that are difficult to read, but please understand.

本書は日本語版説明書を、一般の 日本語→英語 翻訳ソフトで作成したものです。
読みづらい部分も多分にあると思いますが、ご了承ください。

※ The description format of AJATERM · macro internal functions is as follows.

- 1) Arguments enclosed in [XXXXX] are optional. However, it can not be omitted when specifying the following parameters.
ex. `_Func( a [,b] [,c])` -- 'a' must be specified.
'b', 'c' can be omitted, but "b" must also be specified when specifying "c".
- 2) "..." can be specified by repeating the previous parameter
ex. `_Func( a [,b] · · · )` - "b" can be specified repeatedly (many times)
- 3) {XXXXX | YYYYY | ZZZZZ} means to specify either item (XXXXX, YYYYY or ZZZZZ).

1. Option

#	Function format Description	Return value／※Note	Sample
1	option {signed unsigned double} Specify the type of numeric variable. This command must be placed at the beginning of the macro text. signed - We treat numeric variables in macro text as "signed integers" unsigned - We treat numeric variables in macro text as "unsigned integers" double - Handle numeric variable in macro text as "real number"	— The integer is 64 bits Real number is double precision (64bits)	M010
2	_EnableStdRClk(fEnable) Allow / prohibit standard popup menu by right click. If you allow, a standard popup menu will be displayed by right clicking. If prohibited, the right click event becomes effective, and _OnRClick () is executed by right click. fEnable - _TRUE: Standard popup menu permission (default) _FALSE: standard popup menu disabled (_OnRClick () executable) ※Set the behavior when right clicking without pressing Shift/Ctrl key. If you right-click while holding down the A key, a right click event always occurs regardless of this setting	— ※Standard pop-up menu 	M010 M070
3	_EnableAutoScroll(fEnable) Enable / Disable auto scroll (scroll to bottom line of screen at drawing) fEnable - _TRUE : Autoscroll allowed (default) _FALSE : No auto scroll	—	M020

2. Screen information

#	Function format Description	Return value／※Note	Sample
1	_SetVramSize(width, height) Sets the size of the VRAM. The display contents are cleared. width : VRAM width (single byte characters, default = 512 characters) height : VRAM height (number of rows, default = 128 rows) ※It depends on the VRAM size set in the application that uses this control. (The default is maximum, 512 digits, 128 lines)	— ※The cursor can be moved to an arbitrary position in the VRAM by _locate ().	M020 M030
2	_SetBufLines(lines) Sets the number of lines in the scroll buffer. The display contents are cleared. lines : Number of scrollable lines (default is 20,000 lines)	— ※Set number of scrollable lines	M030
3	_GetVRamWidth() Get the VRAM width (single byte characters)	VRAM width ※The default is 512 digits	M030
4	_GetVRamHeight() Get the height (number of lines) of VRAM	VRAM height ※Default is 128 lines	M030
5	_GetBufLines() Get the number of lines of the scroll buffer (including the number of lines of VRAM)	Scroll buffer lines ※Default is 20,000 lines	M030

3. Display

#	Function format Description	Return value／※Note	Sample
1	_Printf(\$fmt [,arg] . . .) Display formatted character string. \$fmt – Format string (almost the same as C printf ()) arg – Argument corresponding to format token (% . . .)	—	M010 M020 M050
2	_Print({item1 \$item1} [, {item2 \$item2} . . .) Multi-item display to cursor position item / \$item - Numeric / String (The item specified by the argument, the numeric value is decimal, the character string is output as it is)	—	M020
3	_Pd({item \$item} [,len] [,lup]) One item display output to the cursor position item / \$item - Numeric / String len – The total number of display digits (left-justified in case of negative number, same as effective character length when omitted) lup – Number of display digits after decimal point (valid only in real number mode)	— ※In real number mode, if "lup" is omitted, the decimal part is always displayed in 6 digits	M061 M070 M101
4	_Ph({item \$item} [,len]) One item display output to cursor position item / \$item - umeric / String len – The total number of display digits (left-justified in case of negative number, same as effective character length when omitted) (The item specified by the argument, the numerical value is hexadecimal, the character string is output as it is)	—	M051 M052 M070
5	_Locate(x, y) Drawing position setting x: Digit position (0 to VRAM width - 1) y: Row position (0 to the number of rows of VRAM - 1)	— ※Normally Disable auto scroll with _EnableAutoScroll (_FALSE)	M020
6	_Cls() Clear screen and drawing color reset (drawing color = black, background color = white)		M020
7	_SetTextColor(color) Sets the drawing color of the text. color: Text color (0: black, 1: red, 2: green, 3: yellow, 4: blue, 5: purple, 6: light blue, 7: white)		M020
8	_SetBackColor(color) Sets the background color of the text. color: background color (0: black, 1: red, 2: green, 3: yellow, 4: blue, 5: purple, 6: light blue, 7: white)		M020
9	_ResetColor() Set the drawing color of text = black, background color = white.		M020

4. Array

#	Function format Description	Return value/※Note	Sample
1	_Elements(arr[]) Get number of elements of array variable	Number of elements of array	M040 M063
2	_MakeArray(n) Returns a numeric array of the number of elements specified by n. (Ex. Arr [] = _MakeArray (10);) ※ ex. n[] = _MakeArray(10) --- Change n [] to an array with 10 elements	※Change the variable that receives the return value to the numerical array of the number of elements n	M040
3	\$_MakeArray(n) Returns a character array of the number of elements specified by n. (Ex. \$ Arr [] = \$_MakeArray (10);) ※ ex. \$s[] = \$_MakeArray(10); - Change \$ s [] to an array with the number of elements = 10	※Change the variable that receives the return value to the character array of the number of elements n	M040

5. Integer operation

#	Function format Description	Return value/※Note	Sample
1	_CalcCrc(knd, bno, loc, len [, ini]) Calculation of 16 bit CRC knd – Operation type _CRC_ITL : CCITT X.25 Left feed _CRC_ITR : CCITT X.25 right feed _CRC_16L : ANSI-CRC Left feed _CRC_16R : ANSI-CRC right feed bno – Memory block number loc - Location of memory block len - Number of bytes ini – Initial value (0x0000 or 0xFFFF)	CRC 値 ※Generator polynomial of CCITT X.25 $X^{16} + X^{12} + X^5 + 1$ ※ Generator polynomial of ANSI-CRC $X^{16} + X^{15} + X^2 + 1$	M021
2	_SRand(n) Initialization of random sequence of pseudo random numbers n = 1: Reinitialization of random number generator n ≠ 1: Initialization of random number sequence	—	M020
3	_Rand() Obtain pseudorandom value	Obtain pseudorandom value (0~32767)	M020

6. Arithmetic operation

#	Function format Description	Return value/※Note	Sample
1	_Sqrt(n) Get square root of n	square root value	M100
2	_Sin(degree) Get sine value (unit in degrees)	sine value	M100 M120
3	_Cos(degree) Get cosine value (unit in degrees)	osine value	M100 M120
4	_Tan(degree) Get tangent value (unit in degrees)	tangent value	M100 M120

5	_ASin(n) Get arc sine value of n	arc sine value (-90~+90)	M100
6	_ACos(n) Get arc osine value of n	arc osine value (0~180)	M100
7	_ATan(n) Get arc tangent value of n	arc tangent value (-90~+90)	M100
8	_Atan2(y, x) Get arc tangent value of y/x	arc tangent value(-180 ~ +180)	M100
9	_Abs(n) Get absolute value of n	absolute value	M100 M120
10	_Min(a, b) Returns the smaller of a and b	smaller of a and b	M120
11	_Max(a, b) Returns the larger of a and b	larger of a and b	M120

7. Timer/Wait

#	Function format Description	Return value/※Note	Sample
1	_CurTime() Get the current time (local time) in total number of seconds since 1970/1/1 00: 00: 00	Total number of seconds from 1970/1/1 00:00:00	M020 M090
2	_GetTime(year, month, day, hour, minute, second) Get the specified date and time as number of seconds from "1970/1/1 00: 00: 00".	Total number of seconds from 1970/1/1 00:00:00	M090
3	_GetDateAndTime([seconds]) Get the date and time from the total number of seconds since 1970/1/1 00: 00: 00 If you omit seconds, you get the current date and time (ex. Today[] = _DateAndTime();)	Array containing date and time [0]: Year [3]: hour [1]: Month [4]: minute [2]: Day [5]: second [6]: millisecond	M020
4	\$_Date([seconds]) Get the date string from the total number of seconds since 1970/1/1 00: 00: 00 If you omit seconds, you get the current date	Date string ("yyyy/mm/dd")	M020 M090
5	\$_Time([seconds]) Get the time string from the total number of seconds since 1970/1/1 00: 00: 00 If you omit seconds, you get the current time	Time string ("hh:mm:ss")	M090
6	\$_DateAndTime([seconds]) Get the time string from the total number of seconds since 1970/1/1 00: 00: 00 If you omit seconds, you get the current date and time	Date and Time string ("yyyy/mm/dd hh:mm:ss")	M020 M090
7	_msWait(msTime) Weight in milliseconds (Measure the passage of time with a sleep (1) loop. The resolution depends on the resolution of the Windows timer (about 10 to 16 ms))	—	M020 M120
8	_usWait(usTime) Weight in microseconds (While waiting, it loops without Sleep () and keeps using the CPU)	—	M090

8. Timer Start / Stop

#	Function format Description	Return value／※Note	Sample
1	_TimerStart(tid, period) Timer event "_OnTimer()" is generated at regular intervals. tid - Timer ID(1 ~31) period - Timer event activation cycle [ms]	—	M062
2	_TimerStop([tid]) Stops generation of timer event. tid - Timer ID(1 ~31) ※If "tid" is omitted, all timers are stopped.	—	M062

9. String

#	Function format Description	Return value／※Note	Sample
1	_StrLen(\$str) Gets the length of the string.	string length.	M061
2	_StrICmp(\$str1, \$str2 [,nEqual]); String comparison (Comparison without distinguishing between uppercase letters and lowercase letters ※1) nEqual: Return value when string matches (Assumes 0 when nEqual is not specified)	= nEqual: \$str1 equal \$str2 > nEqual : \$str1 > \$str2 < nEqual: \$str1 < \$str2	M130
3	_StrStr(\$str1, \$str2) Search substring (\$ str2) in character string 1 (\$ str1) (Compare upper case and lower case letters)	≠ -1: Position of substring (Zero or more) = -1:Not found	M061
4	_StrIstr(\$str1, \$str2) Search substring (\$ str2) in character string 1 (\$ str1) (Compare case without distinguishing between uppercase and lowercase letters)	≠ -1: Position of substring (Zero or more) = -1:Not found	M130
5	_ToValue(\$str) Convert a numeric expression string to a numeric value (If the head is "0x" it is a hexadecimal integer) \$str: Numeric expression string (ex. "123", "-3.14", "0xFF")	Converted number	M130
6	\$_StrCat([\$str1] [, \$str2] . . .) Get string concatenated with all specified character strings	Concatenated string	M060 M070
7	\$_PathCat(\$str1, \$str2) Create a character string in which one "¥" is always inserted between two character strings	Concatenated string	M130
8	\$_StrPart(\$str, loc, len) Get partial string in \$Str loc: Byte position of a substring in \$ str (negative number is also acceptable) len: Length of substring in \$ str (In case of a negative number, the length to the left ex. \$_StrMid(\$ s, 3, -3); First 3 letters)	partial string ※If loc, len are outside the range of the string, assume the right / left end of the mapped string. However, when outside the character string is mapped, an empty string	M070
9	\$_StrRep(\$str1, loc, len, \$Str2) Replace part of the string (\$str1) with \$str2 loc: Byte position of the substring in \$str1 (length of 0 to \$str1) len: The length of the substring (if it is 0, insert the string (\$Str 2))	String with substring replaced ※Correct it to 0 if loc is a negative number and \$ str1 if it exceeds \$ str	M072

10	\$_StrLTrim(\$str) \$_StrLTrim(\$str[]) Remove front space of string \$str : Source string that creates a string with spaces removed \$str[] : String array to remove spaces	String without front spaces ※ You specified an array (\$str[]) If there is no return value.	M130
11	\$_StrRTrim(\$str) \$_StrRTrim(\$str[]) Remove trailing space of string \$str : Source string that creates a string with spaces removed \$str[]:String array to remove spaces	String without trailing spaces ※ You specified an array (\$str []) If there is no return value.	M130
12	\$_StrTrim(\$str) \$_StrTrim(\$str[]) Remove spaces at both ends of the string \$str : Source string that creates a string with spaces removed \$str[]:String array to remove spaces	String without both spaces ※ You specified an array (\$ str []) If there is no return value.	M063
13	\$_StrSplit(\$str [, \$delimiter]) Return an array of character strings obtained by decomposing the character string (\$str) with a delimiter (\$delimiter) \$str : Source string to create an array decomposed by delimiter \$delimiter : Delimiter character string (Comma (",") when not specified)	Array of character strings decomposed by delimiter	M063
14	\$_StrJoin(\$str[] [, \$delimiter]) Each element of the string array (\$str[]) is joined by a delimiter (\$delimiter) to make it a single character string \$str[] : String array for joining elements by delimiter \$delimiter: Delimiter character string (Comma (",") when not specified)	String connected by delimiter	M130
15	\$_Sprintf(\$fmt [, arg] . . .) Format string creation (Convert numeric items to hexadecimal string) \$fmt - Format string (almost the same as C printf ()) arg - Argument corresponding to format token(% . . .)	Formatted string	M130
16	\$_Input([\$Prompt] [, \$DefStr]) Input of character string by dialog (maximum 511 bytes \$Prompt - Title (character string displayed in title bar of input dialog) \$DefStr - Initial character string (empty string when omitted)	Input / selected character string ※To input a character string, select an item from the combo box or directly input a character string.	M061 M063 M070 M072
17	_HexStrToBinary(\$HexStr, MemBlkNo, loc) Two hexadecimal numbers separated by white space are converted into binary data and stored in the memory block. \$HexStr - hexadecimal character string (Ex. "41 42 0 D 0 A") MemBlkNo - Memory block number Loc - byte position (0 to) ※ In the following cases, conversion to binary data is canceled. · Hexadecimal value other than 0x00 to 0xFF · Detects other than hexadecimal characters · Over Memory Block	Number of bytes of binary data	M070
18	_GetValueByKey(\$str, \$key [, fExact]) Numerical acquisition following the key (\$key) in the character string (\$str) \$str - String \$key - Key fExact - Key comparison method (_ TRUE: case sensitive, _ FALSE: do not distinguish) ※ ex. _GetValueByKey("-- X=123 --" , "X=") → Return numeric value 123	Numbers following the key ※ Returns 0 if there is no key or when there is no subsequent value.	—
19	_GetValues(\$str) Returns an array that collects numeric values from the token in the string. \$str - String to extract number ※ ex. v[] = _GetValues("-- x = 123 (r is 10, q is -20)") → v[0] = 123, v[1] = 10, v[2] = -20 ※ Lexical unit is same as C (comments "/" * . * "/" and "// . ." are also valid)	Array of collected numbers ※If there is no numeric value, it returns value = 0 with an array of element number = 1.	—

※ 1 : When distinguishing between uppercase letters and lowercase letters, it can be described by a comparison expression of a character string (ex. \$ Txt == "ABC")

10. Accept

#	Function format Description	Return value／※Note	Sample
1	_MsgBox(\$Message, \$Title, BoxType) Display a message box for inquiries. \$ Message - inquiry message \$ Title - String to display in the title bar of the message box BoxType - type of message box _MB_OK - [OK] Show push buttons only _MB_OKCANCEL - Display OK, Cancel push buttons _MB_RETRYCANCEL - Display push buttons for "Retry" and "Cancel" _MB_YESNO - Show [Yes] and [No] push buttons _MB_YESNOCANCEL - Display Yes, No, Cancel push buttons _MB_ICONERROR - Stop icon icon is displayed.	_IDOK - Select "OK" _IDCANCEL - Select "CANCEL" _IDNO - Select "NO" _IDYEAS - Select "YES"	M070
2	_PopupMenu(\$menu[], x, y [, \$Func]) Displays a popup menu and returns the selected item. \$Menu [] - menu item (character array) x, y - the display position of the menu (the upper left corner of the macro window is the origin (0, 0)) \$Func - an event function to execute when the menu is selected	When \$ Func is not specified ≥ 0 : The selected menu item (Index value of item) -1 : Menu canceled ※! When \$Func is specified, no return value	M010 M070 M071 M072

11. Memory block

#	Function format Description	Return value／※Note	Sample
1	_MemAlloc(BlockSize) Allocate memory block BlockSize – Number of bytes of allocated memory block (1 to 0xFFFFFFFF)	Memory block number	M050
2	_MemFree(MemBlkNo) Releases the memory block. MemBlkNo - Memory block number	—	M050 M060 M070
3	_GetMemBlk(MemBlkNo, loc, len [, fChangeEndian]) Read the data in the memory block. Reads an integer value of the specified number of bytes from the specified position. MemBlkNo - Memory block number loc - Byte position(0 or more) len - Number of bytes fChangeEndian - Convert endian format with _TRUE	The data value read from within the memory block (In real number mode, convert the read integer value (8 to 64 bits) to a real value.)	M050 M051 M052 M070
4	_PutMemBlk(MemBlkNo, loc, len, value [, fChangeEndian]) Write data into memory block Write the integer value of data (lower 8 to 64 bits) to specified byte position, byte number MemBlkNo - Memory block number Loc - byte position (0 to) Len - number of bytes (1 to 8) value - the data value to be written into the memory block In real number mode, convert the real number to an integer value (8 to 64 Bit) and write it fChangeEndian - Convert endian format with _TRUE	—	M050 M051 M052

5	_ClrMemBlk (MemBlkNo, ByteValue) Clear the memory block with byte value. MemBlkNo - Memory block number Value - Byte position(0~255)	—	M050 M051 M052
6	_DumpMemBlk (MemBlkNo [, loc] [, len]) The contents of the memory block are displayed in hexadecimal dump. MemBlkNo - Memory block number Loc - byte position (assuming 0 if omitted) Len - number of bytes (default: assuming the number of bytes to the end of the memory block)	—	M050 M051 M052

12. File / Folder

#	Function format Description	Return value／※Note	Sample
1	_FOpen (\$FilePath, \$Access) Open the file. \$FilePath – File path name \$Access - File access specification ("rt", "wt", "at", "rb", "wb" or "ab") "rt": Read text file "rb" : Read binary file "wt": Write text file "wb": Write binary file "at": Append text file "ab" : Append inary file	File number (1 to 31)	M060 M061 M062
2	_FClose (fno) Close the file. fno - File number (Return value of _FOpen)	— ※When closing the macro, all files which are not closed are automatically closed	M060 M061 M062
3	_FPutS (fno, \$str) Write a character string to the file. fno - File number (Return value of _FOpen) \$str - String to write to file ※ _FOpen file access specification must be "wt" or "at".	—	M060
4	_FRead (fno, bno, loc, len) Read binary data from file. fno - File number (Return value of _FOpen) bno - Memory block number (return value of _MemAlloc) loc - Byte position (location in memory block) where the read data is stored len - Number of bytes to read ※ _FOpen file access specification must be "rb".	Number of bytes read	M060
5	_FWrite (fno, bno, loc, len) Writes binary data to the file. Fno - File number (Return value of _FOpen) bno - memory block number (return value of _MemAlloc) Loc - Byte position of data to be written (location in memory block) Len - number of bytes to write ※ _FOpen file access specification must be "wb" or "ab".	Number of written bytes	M060
6	_FGetSeek (fno) Get the current file pointer (file access byte position). fno - File number (Return value of _FOpen)	Current file pointer	M060
7	_FSetSeek (fno, loc) Sets the file pointer (file access byte position). fno - File number (Return value of _FOpen) loc - File pointer	—	M060

8	\$_FGetS(fno) Reads a character string from a file. (One line, maximum 1023 bytes) fno - File number (Return value of _FOpen) ※ _FOpen file access specification must be "rt".	The read character string	M060 M061 M062
9	_FDelete(\$FilePath) Deletes the specified file. \$FilePath - Path name of the file to delete (ex. 'd:¥work¥sub¥temp¥Sample.txt');	—	M060
10	_MakeFolder(\$FolderPath) Create the folder specified by absolute path name. (End of path name must be "¥") \$FolderPath - Absolute path name of the folder to be created (ex. 'D: ¥ work ¥ sub ¥ temp ¥');	—	M140
11	_PathExists(\$Path) Check if path exists \$Path - Path name to check for existence	_TRUE : Exist _FALSE : Not exist	M140
12	_PathIsFolder(\$Path) It checks whether the specified path is a real folder \$Path - Path name to check if it is a real folder	_TRUE : Existing folder _FALSE : Not existing folder	M140

13. Acquire file name / folder name

#	Function format Description	Return value／※Note	Sample
1	\$_GetOpenFile([\$Title] [, \$Filter [[, \$DefExt] [, \$DefFile]]) Select a file to open by dialog. \$ Title - dialog title (empty string when not specified) \$ Filter - File selection filter ("All (*. *)/*.*) when not specified --- ※ 1 \$ DefExt - default extension ("txt" when not specified) \$ DefFile - Default file / path name (empty string when not specified)	Path name of selected open file (If canceled Empty string (""))	M070
2	\$_GetSaveFile([\$Title] [, \$Filter] [, \$DefExt] [, \$DefFile]) Use the dialog to select / specify the file to be saved. \$ Title - dialog title (empty string when not specified) \$ Filter - File selection filter ("All (*. *)/*.*) when not specified ---※ 1 \$ DefExt - default extension ("txt" when not specified) \$ DefFile - Default file / path name (empty string when not specified)	Path name of selected save file (If canceled Empty string (""))	M061 M062 M071 M072
3	\$_GetFolder([\$WndTitle] [, \$BoxTitle] [, DefFolder]) Select a folder by dialog. \$ WndTitle - Title to be displayed in the window title bar (empty string when not specified) \$ BoxTitle - Title to display in the dialog box (empty string when not specified) \$ DefFolder - default folder / path name (empty string when not specified)	Path name of the selected folder (If canceled Empty string (""))	M063 M072
4	\$_GetMacFilePath() Get the macro file / path name currently being executed.	Path name of macro file	M064
5	\$_SearchFiles(\$Folder, fSubDir [{, \$Wild \$Wild[] }])... It searches for a file and returns an array of found file path names. \$ _Folder - the folder path name to search fSubDir - subfolder search flag (_TRUE: also search subfolders, _FALSE: not search) \$ Wild - Wildcard to search \$ Wild [] - wildcard to search (multiple designation by array) ex. \$ files [] = _SearchFiles ('d: ¥ work', _TRUE, "*. c", "*. h");	Array of found file path names ※In the macro text If there is a "_OnFileSearch ()" function, it calls this and notifies the name of the folder being searched. (See the event processing function)	M063

6	\$_SplitPath(\$PathName) Path name decomposition (decomposes an absolute path name and returns a character type array with 4 elements) ex. <code>\$s[] = \$_SplitPath('c:\sub_dir\sample.txt');</code>	String array of the following contents [0]:Drive (ex. 'c:') [1]:FolderPath (ex. '\sub_dir\') [2]:FileName (ex. 'sample') [3]:Extension (ex. '.txt')	M064
7	\$MakePath(\$drive [, \$dir] [, \$fname] [, \$ext]) \$MakePath(\$parts[]) Path name creation (creation of path name linking drive, folder name, file name, extension) \$drive: Drive (ex. 'c:') \$dir : FolderName (ex. '\sub_dir\') \$fname: FileName (ex. 'sample') \$ext : Extension (ex. '.txt') \$ parts [] - In a string array of 4 items, [0] = drive , [1] = folder name, [2] = file name, [3] = extension	Concatenated absolute path name ex. 'c:\sub_dir\sample.txt'	M064

※ 1 : Specify a pair of "title" and "wild card" separated by a slash (/).

For example, to specify a text file and all files, specify "TextFile (*. Txt) / *. Txt / All Files (*. *) /*.*"

14. Profile

#	Function format Description	Return value／※Note	Sample
1	_SetProfileSect(\$sect) Set section of profile to access \$sect - Section name The default section name when this function is not executed is "DefSect".	—	M061
2	_SetProfileVal(\$key, value) Record the value to the profile section. \$key - Key name value - Value to record in profile	—	M150
3	_SetProfileStr(\$key, \$string) Record the character string in the profile section. \$key - Key name \$string - String to record in profile	—	M061 M063
4	_GetProfileVal(\$key, DefVal) Read values from profile section. \$key - Key name DefVal - Default value	Value read from profile If the key does not exist, it returns DefVal.	M150
5	\$GetProfileStr(\$key, \$DefStr) Read a character string from the profile section \$key - Key name \$DefStr - Default string	String read from profile If the key does not exist, \$ DefStr is returned.	M061 M063

15. Com-port / Mail-slot

#	Function format Description	Return value / ※Note	Sample
1	_ComCreate ([PortNo \$MySlotName]) Initialize the COM port. (I will not open it yet) PortNo - COM Port Number (1 to 256) \$MySlotName - Your mail slot name (when creating mailslot) ※Unspecified port information is read from the profile. ※If no argument is specified, the previous setting value is adopted.	Port number (1 to 256)	M062 M070 M071 M073 M074 M080
2	_ComCreateByDialog ([OldPort]) By setting dialog parameters, set the port parameters and initialize the port. OldPort - Old port number ※If you specify the same port in the open state and press the OK button, reopen the port concerned. ※When creating a new port, if "OldPort" is specified, this old port will be discarded (if new and old port numbers are different). However, if OldPort = 0, do not do anything ※If you select a different port without specifying "OldPort", the old port will not be destroyed and will remain intact.	1 to 256 - Generated port number 0 - Cancel / Error	M070 M071 M072 M080
3	_ComDelete (PortNo) Discard the port initialized with _ComCreate / _ComCreateByDialog. If the port is open, close it and discard it. PortNo - Port number (1 to 256) ※Port information is recorded in the profile.	— ※ At the end of macro execution, ports not destroyed are automatically discarded.	M070 M071 M072 M080
4	_ComOpen (PortNo [, Rate] [, DataLen] [, Parity] [, StopBit]) _ComOpen ([RemoteSlotName] [, RemoteHostName]) _ComOpen (256) 1) Open the COM port. PortNo - COM port number (1 to 255) Rate - Communication rate [bps] DataLen - Data length (7 to 8) Parity - Parity bit (0: None, 1: Odd number, 2: Even number) StopBit - Stop bit length (1 to 2) 2) Open mailslot RemoteSlotName - Name of the other mailslot RemoteHostName - name of the other computer (your computer when empty string ("") is specified) 3) Open mailslot with current value (specify port = 256)	— ※ The port is _ComCreate / _ComCreateByDialog It must be initialized by. ※ When no argument is specified, the previous value is adopted.	M062 M070 M071 M072 M080
5	_ComIsOpenPossible (PortNo) Check if the port is openable. PortNo - Port number (1 to 256)	_TRUE - Openable _FALSE - Impossible to open	M070 M071 M080
6	_ComIsOpened (PortNo) Check if the port is open. PortNo - Port number (1 to 256)	_TRUE - Opened _FALSE - Closed	M070
7	_ComClose (PortNo) Close the port. If the port is already closed, do not do anything. PortNo - Port number (1 to 256)	—	M062 M070 M071 M072 M080

8	<u>_ComSendText</u> (PortNo, \$Text1 [, \$Text2]) ...) Send text data to the port. PortNo - Port number (1 to 256) Text 1 - text to send	—	M062 M070
9	<u>_ComSendBinary</u> (PortNo, MemBlkNo, loc, len) Send binary data to the port. PortNo - Port number (1 to 256) MemBlkNo - Memory block number Loc - Byte position of transmission data (location in memory block) Len - number of bytes of transmission data	—	M070
10	<u>_ComSendBytes</u> (PortNo, b1 [, b2] ...) Send byte data (multiple bytes possible) to the port. PortNo - COM port number (1 to 256) B1 [, b2] ... - transmission byte data group	—	M070
11	<u>_ComSendPacket</u> (PortNo, MemBlkNo, loc, len) Send the packet data to the port. PortNo - Port number (1 to 256) MemBlkNo - Memory block number Loc - Byte position of transmission data (location in memory block) Len - number of bytes of transmission data	— ※Sending binary data sandwiched between DLE · STX and DLE · ETX	M070
12	<u>_ComGetRate</u> (PortNo) Get the set value of COM port (transfer rate [bps]) PortNo - Port number (1 to 255)	Transfer speed	M070
13	<u>_ComGetDataBits</u> (PortNo) Get the setting value (data bit number) of COM port PortNo - Port number (1 to 255)	7 ~ 8	M070
14	<u>_ComGetParitely</u> (PortNo) Get the setting value (parity bit) of COM port PortNo - Port number (1 to 255)	0 : なし 1 : 奇数パリティ 2 : 偶数パリティ	M070
15	<u>_ComGetStopBits</u> (PortNo) Get the setting value (number of stop bits) of COM port PortNo - Port number (1 to 255)	1 ~ 2	M070
16	<u>\$_ComGetParam</u> (PortNo) Get port setting value as a character string PortNo - Port number (1 to 256) ※In case of mail slot (PortNo = 256), it returns the path name of the destination slot.	“COMn, <r>, <d>, <p>, <s>” <r>:Baud rate[bps] <d>:Data bits(7 or 8) <p>:Parity(No, Odd or Even) <s>:Stop bits(1 or 2)	M070 M071 M080
17	<u>_ComSetDTR</u> (PortNo, fDTR) Set the COM port DTR signal PortNo - Port number (1 to 255) fDTR - DTR signal setting value (_TRUE: valid, _FALSE: invalid)	—	M080
18	<u>_ComSetRTS</u> (PortNo, fRTS) Set the COM port DTR signal PortNo - Port number (1 to 255) fRTS - RTS signal setting value (_TRUE: valid, _FALSE: invalid)	—	M080
19	<u>_ComGetSignal</u> (PortNo) Get the signal state of the COM port. PortNo - Port number (1 to 256)	Bit4(0x10) - CTS signal status Bit5(0x20) - DSR signal status Bit6(0x40) - RING signal status Bit7(0x80) - RLSD signal status Both are "1" and valid state	M080

※Port number = 256 means mailslot

COM-port / Mail-slot event registration

#	Function format Description	Return value / ※Note	Sample
1	_ComSetOnRxText (PortNo, \$func) Register event processing function when receiving text from port PortNo - Port number (1 to 256) \$ Func - function name to execute when receiving text	—	M070
2	_ComSetOnRxCtrl (PortNo, \$func) Register event processing function when receiving control code from port PortNo - Port number (1 to 256) \$ Func - Function name to execute when receiving control code	—	M070
3	_ComSetOnRxEsc (PortNo, \$func) Register event processing function when receiving control code from port PortNo - Port number (1 to 256) \$ Func - Function name to execute when receiving control code...	—	M070
4	_ComSetOnRxPacket (PortNo, \$func, MemBlkNo) Register event processing function when receiving packet data from port PortNo - Port number (1 to 256) \$ Func - Function name to be executed when packet data is received MemBlkNo - memory block number for storing packet data	—	M070

16. COM-port / Mail-slot event processing function

#	Function format Description	Return value / ※Note	Sample
1	OnRxText(\$Text) Processing when text is received from the port \$Text - the received text data	— ※The actual function name is It becomes the function name registered with _ComSetOnRxText ().	M070
2	OnRxCtrl(Ctrl) Perform processing when receiving control code from port Ctrl - the received control code	— ※The actual function name is It becomes the function name registered with _ComSetOnRxCtrl ().	M070
3	OnRxEsc(\$Esc) Processing when ESC sequence character string from port is received \$Esc - the received ESC sequence string	— ※The actual function name is It becomes the function name registered with _ComSetOnRxEsc ().	M070
4	OnRxPacket(len) Perform processing when receiving packet data from the port Len - the number of bytes of received packet data ※Received packet data is stored from the beginning of the memory block specified by _ComSetOnRxPacket ().	— ※The actual function name is It becomes the function name registered with _ComSetOnRxPacket ().	M070

※The actual function name is specified by COM port / mail slot event registration group "_ComSetOnXXXXX ()".

※The above event can be disabled / allowed by _DisComEvt () / _EnaComEvt ().

17. File transfer by XMODEM / YMODEM

#	Function format Description	Return value／※Note	Sample
1	<u>_ComSendXModemSum</u> (PortNo, \$File) - 128Bytes Block, Check sum <u>_ComSendXModemCrc</u> (PortNo, \$File) - 128Bytes Block, CRC <u>_ComSendXModem1K</u> (PortNo, \$File) - 1KBytes Block, CRC Send files by XMODEM protocol. PortNo - Port number (1 to 256) \$ File - the path name of the file to send	— ※ Log status display during transmission / reception ‘.’ - 1024 bytes data Tx / Rx ‘.’ - 128 Bytes data Tx / Rx ‘R’ - retry occurrence ‘E’ - End transmission ‘C [SUM / CRC / 1K]’ – protocol change	M071 M073
2	<u>_ComRecvXModemSum</u> (PortNo, \$File) - 128Bytes Block, Check sum <u>_ComRecvXModemCrc</u> (PortNo, \$File) - 128Bytes Block, CRC <u>_ComRecvXModem1K</u> (PortNo, \$File) - 1KBytes Block, CRC Receive files by XMODEM protocol. PortNo - Port number (1 to 256) \$ File - Path name of the file that stores received data	— ※ Log status display during transmission / reception ‘.’ - 1024 bytes data Tx / Rx ‘.’ - 128 Bytes data Tx / Rx ‘R’ - retry occurrence ‘E’ - End transmission ‘C [SUM / CRC / 1K]’ – protocol change	M071 M074
3	<u>_ComSendYModem</u> (PortNo, \$Wild [, fSubDir] [, fFolder]) Send files by YMODEM protocol. PortNo - Port number (1 to 256) \$ Wild - Wild path name of the file to send FSubDir - subdirectory search flag (_TRUE: search, _FALSE: not search) fFolder - a flag indicating whether to add path information to the transmission file name _TRUE: Add path information to the file name to be transmitted _FALSE: Do not add path information to the file name to be sent	— ※ Log status display during transmission / reception ‘.’ - 1024 bytes data Tx / Rx ‘.’ - 128 Bytes data Tx / Rx ‘R’ - retry occurrence ‘E’ - End transmission	M072 M075 M076 M077
4	<u>_ComRecvYModem</u> (PortNo, \$folder, [, fSubDir]) Receive files by YMODEM protocol. PortNo - Port number (1 to 256) \$ Folder - Path name of the first folder to store received files fSubDir - Operation when path information is added to received file name _TRUE: Create subfolders relative to the first folder _FALSE: All subfolders are not created and stored in the first folder	— ※ : When adding a folder path to a file name, it is a relative path from the first folder (ex. .¥SubDir¥ sample.txt)	M072 M078 M079

18. Standard event processing function

#	Function format Description	Return value／※Note	Sample
1	<u>_OnKeyIn</u> (key) It is executed when key is entered. Key - the key code you entered	— ※Following keys are not notified. CTRL + A(0x01): Select all text CTRL+C(0x03): Copy Selected text	M070
2	<u>_OnRClick</u> (x, y, shift, ctrl) It is executed when right clicking on the macro window. X - the clicked X position Y - the clicked Y position shift - Shift key pressed state (1: pressed, 0: not pressed) ctrl - pressed state of Ctrl key (1: pressed, 0: not pressed)	— ※ x, y are pixel positions with the upper left corner of the macro window as the origin (0, 0)	M010 M070 M071 M072
3	<u>_OnTmcClose</u> () Notify that the time chart window has been closed.	— Called when TmcOpen () is executed and the window is destroyed.	M120
4	<u>_OnButton</u> (BtnNo) Notify that the button has been pressed. BtnNo - Button identification number (0 to 7)	—	M020 M070 M110

5	<u>_OnChkBox</u> (ChkBoxNo, staus) Notify that the status of the checkbox has been set. status - Check status (_TRUE: checked, _FALSE: unchecked) ※Even if the check state is set, even if the check state does not change, it will be executed.	— ※ From within this function, _ChkBoxSetSts () Can not be executed.	M110
6	<u>_OnFileSearch</u> (\$folder) Notify the folder / path name during file search. \$Folder - Folder / path name	_ TRUE: Continue file search _ FALSE: Cancel file search ※ Called while executing \$ _SearchFiles ().	M063
7	<u>_OnTimer</u> (tid) Notify of timer elapsed tid - Timer identification ID (1 to 31)	—	M062
8	<u>_OnMenu</u> (id) Notify that an item has been selected in the popup menu. id - menu item index (0 or more)	— ※The actual function name is specified by the \$Func argument of _PopupMenu ().	M551

※The above function is defined by the user with a fixed function name in the macro text.

※The above events can be disabled / enabled by _DisStdEvt () / _ EnaStdEvt ().

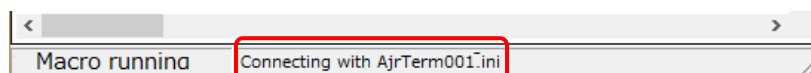
19. AJATERM connection event processing function

#	Function format Description	Return value／※Note	Sample
1	<u>_OnTermLink</u> (\$MySlot, \$RmtSlot) Notify that AJATERM connection is possible.	—	M200
2	<u>_OnTermUnlink</u> () Notify that AJATERM has ended.	—	M200
3	<u>_OnTermOpened</u> () Notify that the serial line on AJATERM is connected.	—	M200
4	<u>_OnTermClosed</u> () Notify that the serial line on AJATERM has been disconnected.	—	M200

※ 1 : To connect with AJATERM, process with _OnTermLink () function as follows. (See sample macro "M200.txt")

```
function _OnTermLink($MySlot, $RmtSlot)
{
    // Mail slot / port generation for connection with terminal
    _ComCreate($MySlot);
    // Receive event function setting
    _ComSetOnRxText (_SLOT_PORT, "OnText");
    _ComSetOnRxCtrl (_SLOT_PORT, "OnCtrl");
    _ComSetOnRxEsc (_SLOT_PORT, "OnEsc");
    _ComSetOnRxPacket (_SLOT_PORT, "OnPacket", bno);
    // Mail slot / port open for connection with terminal
    _ComOpen($RmtSlot);
    . . . . .
}
```

If you connect normally, "Connection with AjrTermnnn.ini" is displayed at the bottom of the window.



(AjrTermnnn.ini is what is displayed in the AJATERM window title)

20. Enable / Disable Event

#	Function format Description	Return value／※Note	Sample
1	<code>_DisStdEvt()</code> Prohibits the execution of standard events. (※1)	— ※ This function is a standard event function (<code>_OnXXX</code>) can not be called.	—
2	<code>_EnaStdEvt()</code> Allows execution of standard events. (※1)		—
3	<code>_GetDisStdEvt()</code> Get the nesting level of prohibiting standard events.	Nesting level of standard event =0 : Enabled >0 : Disabled	—
4	<code>_DisComEvt()</code> Disable execution of COM port communication event. (※1)	— ※ ㇏ The function of the standard event function (<code>_OnXXX</code>) can not be called.	—
5	<code>_EnaComEvt()</code> COM port Allows execution of communication events. (※1)		—
6	<code>_GetDisComEvt()</code> COM port Get the nesting level of communication event disabled.	COM port communication event prohibited nesting level =0 : Enabled >0 : Disabled	—

※1 : Multiple control is prohibited / permitted for events.

That is, execution of the event is enabled when `_EnaXXxEvt()` is executed the same number of times as `_DisXXxEvt()`.

21. Button

#	Function format Description	Return value／※Note	Sample
1	_BtnOpen(BtnNo, \$Caption) Display the button and enable it. BtnNo - Button identification number (0 to 7) \$ Caption - button face string	— ※The buttons are displayed at the top of the control window.	M020
2	_BtnClose(BtnNo) Delete the button BtnNo - Button identification number (0 to 7)	—	M020
3	_BtnEnable(BtnNo, fEnable) Activate / deactivate button BtnNo - Button identification number (0 to 7) fEnable - _ TRUE: Activate (enable) the button _FALSE: Invalidate button (putting it in a non-pressable state)	—	M020

22. Check-Box

#	Function format Description	Return value／※Note	Sample
1	_ChkBoxOpen(ChkBoxNo, \$Caption) Display a check box and enable it. ChkBoxNo - Checkbox identification number (0 to 7) \$ Caption - checkbox / caption string	— ※A check box is displayed at the top of the control window.	M010 M070 M110
2	_ChkBoxClose(ChkBoxNo) Delete the checkbox ChkBoxNo - Checkbox identification number (0 to 7)	—	M160
3	_ChkBoxGetSts(ChkBoxNo) Get checked status of checkbox ChkBoxNo - Checkbox identification number (0 to 7)	Check status _TRUE: Checking _FALSE: Not checked	M160
4	_ChkBoxSetSts(ChkBoxNo, Status) Set the checked state of the check box ChkBoxNo - Checkbox identification number (0 to 7) Status - Check status to be set (_ TRUE: check. _ FALSE: not checked)	—	M110
5	_ChkBoxEnable(ChkBoxNo, fEnable) Enable / disable the check box ChkBoxNo - Checkbox identification number (0 to 7) fEnable - _ TRUE: Enable check box (makes checkable) _FALSE: Invalidate the check box (make it uncheckable)	—	M160

23. Timechart graph

#	Function format Description	Return value/※Note	Sample
1	_TmcOpen([\$Title] [, \$Poroparty] [, x] [, y] [, cx] [, cy]) Open the time chart window \$ Title - Window Title \$ Property - property string (※1) x, y - window display position cx, cy - Window size	— ※If already opened Once the window is closed Open again from.	M120
2	_TmcSetItems(NumOfDataItems) Set number of data items in time chart · graph NumOfDataItems - number of data items (1 to 8)	—	M120
3	_TmcSetRange(Low, High) Range setting of time chart · graph Low - Graph low value High - Graph high value ※To automatically adjust the range of the graph, set _TmcSetRange (0, 0);	— ※It is OK even when Low> High ※When Low = High, the graph Automatically adjust the range	M120
4	_TmcPutData(d1 [, d2] · · · [, d8]) _TmcPutData(data[]) Data administration to time chart · graph D 1 · · · d 8 - data to be administered to the graph data [] - Array of data to be administered to the graph ([1]: d 1 ... [8]: d 8)	— ※Drawing colors as follows d1:  d5:  d2:  d6:  d3:  d7:  d4:  d8: 	M120
5	_TmcClear() Clear time chart graph	—	M120
6	_TmcWndMove(x, y) Move time chart window x, y - screen coordinates of the destination	—	M120
7	_TmcWndSize(cx, cy) Time chart · Wind Size Setting cx, cy - Size of window to set	—	M120
8	_TmcClose() Close the time chart window	—	M120

※1 : Specify the following parameter string. (Empty string if nothing is specified)

P: [L=fff], [H=fff], [B=nnn], [I=nnn], [A=nnn], [BC=nnn], [0=[nnn]/[nnn]], · · · [7=[nnn]/[nnn]]

The beginning of the string must be "P:". (You can put a blank immediately after "P:")

"Fff" is a real number, and "nnn" is an integer (in the case of a hexadecimal number, it is prefixed with '0x').

Separate each parameter with a comma (.). (You can put a space before and after the comma.)

The setting contents of each parameter are as follows. (The order in which each parameter is specified is arbitrary.)

Key word	Content
L	Low rank graph range
H	High order graph range
B	Buffer capacity (number of data stored in buffer)
I	Number of valid data items (1 to 8)
A	Average number
BC	Display color of control outer frame
0 to 7	Specify information of each data item in the form of "offset value / display color"

The display color is specified in hexadecimal "0xbbggrr" format

(bb: blue component, gg: green component, rr: red component)

24. Others

#	Function format Description	Return value／※Note	Sample
1	_ShowText (\$Text) Display text on the upper right of the control window \$ Text - the text string to display	—	M070 M071 M080
2	_GetWndPosX() Get the left end position (screen coordinates) of the control window.	Left end position of control window	M160
3	_GetWndPosY() Get the top edge position (screen coordinates) of the control window.	Top position of control window	M160
4	_GetWndWidth() Gets the width of the control window	Control window width (Number of pixels)	M160
5	_GetWndHeight() Gets the height of the control window	Control window height (Number of pixels)	M160
6	_GetCharWidth() Get character width	Width of single-byte character (Average number of pixels)	M160
7	_GetCharHeight() Get the height of the character	Character height (Number of pixels)	M160
8	_GetLineHeight() Get line height	Line height (Number of pixels)	M160
9	_Exit ([\$Text]) End macro execution \$ Text - This text can be obtained from the parent window with AjcCipGetExitText (). ※The text specified with _Exit () has no effect in the language. The handling of text depends on the application that uses this control. For example, you can use Exit () to specify the macro text file to be executed next.	—	M071

25. Debug

#	Function format Description	Return value/※Note	Sample
1	_VarDump([MaxArrNum]) Display all variable information in the trace display window MaxArrNum - the maximum display item number of the array (assumed to be 10 when not specified)	—	—
2	_Trace (\$fmt [,arg] . . .) Display format string in the trace display window. \$ Fmt - format string (almost the same as p language in p language) Arg - Argument corresponding to format (% . . .)	—	—
3	_Pause() ※ 1 Pause all execution. ※All main processing and event processing are stopped. ※When the step trace window is not opened, a message box is displayed, Press the OK button to resume execution. ※When the step trace window is open, it is displayed that it stopped, Resume with the execution continuation button ().	—	—
4	_Stop() ※ 1 Pause the execution of main processing. ※Only the main processing is stopped. The event will be executed. ※Execution of _Restart () resumes main processing.	—	M010 M070 M080
5	_Restart () Resume main processing which was interrupted by _Stop ().	—	M010 M070 M080
6	_StepTrace(fEnable) Enable / disable display of step trace. fEnable: _TRUE - Display of step trace is allowed _FALSE - display of step trace is prohibited	— ※If the step-trace window is not open, do not do anything.	—

※ 1 : _Pause () and _Stop () stop at the next step of the step including _Pause () / _ Stop () (_Stop () is the next main code step).
(_Pause () / _ Stop () also returns a numeric function (it simply returns 0), so it stops after the formula containing
_Pause () / _ Stop () is executed)

26. Sample macro text list

Macro text	content	Note
M010. txt	Right click and popup menu, execute user event	
M020. txt	Locate & print	
M030. txt	Screen information	
M040. txt	Array operation	
M050. txt	Each operation of the memory block	
M051. txt	Memory block (integer) access	
M052. txt	Memory block (real number) access	
M060. txt	File copy & verify	
M061. txt	Read the text file and display the line containing the specified character string	
M062. txt	Read the text file and send it to COM port at speed of 100 ms / line	
M063. txt	Search files under specified folder	
M064. txt	Disassembly and assembly of file path	
M070. txt	Serial transmission / reception and execution of events	
M071. txt	XMODEM send and receive	
M072. txt	YMODEM send and receive	
M073. txt	XMODEM transmission	
M074. txt	XMODEM reception	
M075. txt	YMODEM transmission (all files under folder)	
M076. txt	YMODEM transmission (All ".C" files under the folder)	
M077. txt	YMODEM transmission (single file)	
M078. txt	Receive YMODEM (Do not create subfolders)	
M079. txt	Receive YMODEM (Create subfolder)	
M080. txt	COM port signal setting and reading	
M090. txt	Time, Weight	
M100. txt	Arithmetic operation	
M101. txt	Find the center of a circle from 3 points	
M110. txt	Buttons and check boxes	
M120. txt	Time chart	
M130. txt	String manipulation	
M140. txt	Folder creation, path check	
M150. txt	Read / write profile (number)	
M160. txt	Generate / delete check box	
M200. txt	Connect with AJATERM and communicate	
M500. txt	Button and check box, COM transmission	Explanation at AjmMacroE.pdf It is a macro that is doing
M510. txt	Send and receive text data via serial port	
M520. txt	Send and receive binary packet data via serial port	
M530. txt	Port state persistence, text / binary transmission and reception	
M550. txt	Right click and popup menu	
M551. txt	Right click and popup menu (improved version)	
M560. txt	Eight sine waves with different phases are displayed	
M570. txt	Text transmission at the current time every 1 second (event driven)	
M571. txt	Text transmission at the current time every 1 second (event-driven not used)	