AVR-DOS Function List

This document contains a list of all functions included in AVR-DOS with their Input and Output parameters. The list is organized by topics.

High level

1) File System
2) File i/O with file name
3) File I/O with file#
4) File I/O with file handle
5) File in directory
6) Sub-Directory

Low-Level

7) File open, create, close, delete
8) Cluster
9) Sector
10) Directory
11) Error-, DIR-, FAT- and file buffer status
12) File handle variable
13) Internal functions

Used Abbreviations:

A: registers r16 for 1 Byte, r16-17 for Word (2 Bytes) , r16-19 for Long (4 Bytes)

B: registers r20 for 1 Byte, r20-21 for Word (2 Bytes) , r20-23 for Long (4 Bytes)

X: X-Pointer addresses value

Z: Z-Pointer addresses value

Y0/1: Softstack addressed by Y+0 and Y+1 contains current File handle address

1) File System

Routine & Label	Input-Setting / Output										
			Input Variable					Function Output		Error	
default Register										r25	
_AVRDOSVER								r24			Returns the AVR-DOS Version
_GetFileSystem			r24 *)							o	Read file system form drive and initialize SRAM-variables of AVR-DOS *) Partition# (1-based)
_GetDiskFreeSize								A		o	Get free capacity on Drive in Kbytes (1024 Bytes)
_GetDiskSize								A			Get capacity of Drive in Kbytes (1024 Bytes)
_FilesAllFlush										o	Write current content of all opened files in SRAM buffer to drive and updates FAT and Directory

2) File I/O with file-name

Routine & Label	Input-Setting: File Name	Input-Setting: File#	Input-Setting: Input Variable	Input-Setting: Length of read / write	Input-Setting:	Input-Setting: Pointer for Output	Output: File#	Output:	Output:	Output: Error	
default Register	X	r24					r24			r25	
_FileOpen_Input	o	o								o	Open file for Input (reading)
_FileOpen_Input0	o						o			o	
_FileOpen_Output	o	o								o	Open file for Output (Writing) and delete existing file
_FileOpen_Output0	o						o			o	
_FileOpen_Append	o	o								o	Open file for Output (writing) to end of existing file (append) and create new file if it doesn't exists
_FileOpenAppend0	o						o			o	
_FileOpen_Binary	o	o								o	Open file for binary read and write and create new one, if it doesn't exists
_FileOpen_Binary0	o						o			o	
_FileOpen_FileNotExists	Z	o	r25[1)]								Is called from _FileOpen_Output, _FileOpen_Append, _FileOpen_Binary if file doesn't exists in directory [1)] file open mode
_BLoad	o					Z				o	Load the content of a file to SRAM
_BSave	o		Z	B						o	Save a SRAM Range to a File

3) File I/O with file#

Routine & Label	Input-Setting: File Name	Input-Setting: File#	Input-Setting: Input Variable	Input-Setting: Length of read / write	Input-Setting: File Pointer Position	Input-Setting: Pointer for Output	Output: File#	Output: Function Output	Output:	Output: Error	
default Register	X	r24			A		r24			r25	
_GetFreeFileNumber							o			o	Get a free file number
_FileLoc		o				X				o	Location of last read or written Byte in File (1-based)
_FileLOF		o				X				o	Length of opened file
_FileSeek		o				X				o	Location of next Byte to read or write in File (1-based)
_FileSeekSet		o			X					o	Set Location of next Byte to read or write in File (1-based)
_FileAttr		o						r24		o	Get File Access (Opening) mode
_FileEOF		o						r24		o	Get End of File Status of an opened File
_FileByteInput		o						r24		o	Read one Byte from the file
_FileByteOutput		o	r25							o	Write one Byte to the file
_FileLineOutput		o	X							o	Write a String to the File and appends a CR+LF
_FileLineInput		o		r25		X				o	Read a Line from the File to a String variable
_FilePutRange		o	X	B	o[T)]					o	Write Binary Data to a file (1 to 65535 Bytes)
_FilePutRange_1		o	X	(1)	o[T)]					o	Byte
_FilePutRange_2		o	X	(2)	o[T)]					o	Word/Integer
_FilePutRange_4		o	X	(4)	o[T)]					o	Long/Single
_FilePutRange_8		o	X	(8)	o[T)]					o	Double
_FilePutRange_Bytes		o	X	r20	o[T)]					o	String (1 to 255 Bytes)
_FileGetRange		o		B	o[T)]	X				o	Read Binary Data from a file (1 to 65535 Bytes)

3) File I/O with file#

Routine & Label	Input-Setting: File Name	Input-Setting: File#	Input-Setting: Input Variable	Input-Setting: Length of read / write	Input-Setting: File Pointer Position	Output: Pointer for Output	Output: File#	Output: Function Output	Output:	Output: Error	
default Register	X	r24			A		r24			r25	
_FileGetRange_1		o		(1)	o$^{T)}$	X				o	Byte
_FileGetRange_2		o		(2)	o$^{T)}$	X				o	Word/Integer
_FileGetRange_4		o		(4)	o$^{T)}$	X				o	Long/Single
_FileGetRange_8		o		(8)	o$^{T)}$	X				o	Double
_FileGetRange_Bytes		o		r20	o$^{T)}$	X				o	String (1 to 255 Bytes)
_FileFlush		o								o	Write current content of file in SRAM buffer to the drive
_FileClose		o								o	Close an opened file

o$^{T)}$ Consider new file-position in registers A (r16-r19) if T-Flag set, otherwise read/write from/to current position

(x) Length is set in routine to value in brackets

4) File I/O with file-handle

Routine & Label	Input-Setting / Output										
	File Handle		Input Variable	Length of read / write	File Pointer Position	Pointer for Output	File#	Function Output		Error	
default Register	Y0/1				A		r24			r25	
_FileWriteByte	o		r24							o	Write one Byte to the File
_FileWriteString	o		X							o	Write a String to the File
_FileWriteStringQuotationMark	o		X							o	Write a String to the File surrounded by quotation mark to the file
_FileWriteCRLF	o									o	Write a Carriage Return (0D) and a Line Feed (0A) to the File
_FileWriteComma	o									o	Write a Comma "," to the File
_FileWriteQuotationMark	o									o	Write a Quotation Mark (") to the File
_FileWriteStrConst	o		Z							o	Write a Constant from the Flash to the File
_FileWriteDecByte	o		Z							o	Write the Numeric of a Byte in ASCII to the File
_FileWriteDecWord	o		Z							o	Write the Numeric of a Word in ASCII to the File
_FileWriteDecInt	o		Z							o	Write the Numeric of a Integer in ASCII to the File
_FileWriteDecLong	o		Z							o	Write the Numeric of a Long in ASCII to the File
_FileWriteDecSingle	o		Z							o	Write the Numeric of a Single in ASCII to the File
_FileWriteDecDouble	o		Z							o	Write the Numeric of a Doublee in ASCII to the File
_FileWrite_StrBuf	o									o	Write the ___StrBuf to the file (string buffer in frame-space)
_FileReadByte	o							r24		o	Read one Byte from the File
_FileReadString	o			r20		X				o	Read a String from the File to a SRAM-String variable and checks for Quotation marks
_FileReadString2StrBuf	o			r20						o	Read a String from the file to StringBuffer (Frame space)

4) File I/O with file-handle

Routine & Label	Input-Setting: File Handle		Input Variable	Length of read / write	File Pointer Position	Output: Pointer for Output	File#	Function Output		Error	
default Register	Y0/1				A		r24			r25	
_FileReadDecNum	o							A*)		o	Read a Number in ASCII from the file and convert it to binary *) BYTE: r16; WORD, INTEGER: r16,r17; LONG: r16-r19
_FileReadDecSingle	o							FP-ACC1 *)		o	Read a Single in ASCII from the file and convert it to binary *) FP-ACC1 = Floating Point Accu 1 = r13-r16
_FileReadDecDouble	o					Z				o	Read a Double in ASCII from the file and convert it to binary
_FileReadByteArray	o			B		X				o	Read a range of Bytes from the file
_FileWriteByteArray	o		X	B						o	Write a range of Bytes to the File
_GotoFilePosition	o				o					o	Goto new position in file (0-based)
_GotoFilePosition_1Base	o				o					o	Goto new position in file (1-based)

5) File in directory

Routine & Label	Input-Setting: File Name		Input-Setting: Input Variable			Output: Pointer for Output				Output: Error	
default Register	X				A					r25	
_FileRename	Z[1]		X[2]							o	Rename a file in current directory [1] Old file-name [2] New file-name
_Dir	o					Z				o	Search a file name on the drive
_Dir0						Z				o	Search next file name on the drive
_FileLen	o					Z				o	Get length of file on the drive
_FileLen0						Z				o	Get Length of last searched (DIR) file
_FileDateTimeB	o					Z				o	Get Date&time as 6-byte-array of file on the drive
_FileDateTimeB0						Z				o	Get Date&time as 6-byte array of last searched (DIR) file
_FileDateTimeS	o					Z				o	Get Date&time as String of file on the drive
_FileDateTimeS0						Z				o	Get Date&time as string of last searched (DIR) file
_FileDateS	o					Z				o	Get Date as String of file on the drive
_FileDateS0						Z				o	Get Date as string of last searched (DIR) file
_FileTimeS	o					Z				o	Get Time as String of file on the drive
_FileTimeS0						Z				o	Get Time as string of last searched (DIR) file
_GetAttr	o					Z				o	Get DOS-attribut of file on the drive
_GetAttr0						Z				o	Get DOS-attribut of last searched (DIR) file

6) sub directory

Routine & Label	Input-Setting: File Name	File#	Input Variable	Length of read / write	File Pointer Position	Pointer for Output	Output: File#	Function Output		Error	
default Register	X	r24			A		r24			r25	
_MKDir	X									o	Creates a new Sub-Directory
_ChDir	X									o	Changes to another directory (sub-directory, upper or root directory)
_RmDir	X									o	Removes a directory

7) File open, create, close, delete

Routine & Label	Input-Setting: File#	Input-Setting: File Handle	Input-Setting: Dir Enrtry#	Input-Setting: File Name	Input-Setting: File Open Mode	Input-Setting: Pointer for Output	Output: File Handle	Output:	Output:	Output: Error	
default Register	R24	Y0/1	B	Z	r25		Z			r25	
_SetFileHandle	o		o		o		o			o [E1]	Set a file handle in file handle buffer [*)] File open mode
_GetFileHandle	o						o			o [E2]	Get a file handle
_ClearFileHandle		Z									Clear the variable range of a file handle
_ClearFileHandle_Y		o									
_CloseFileHandle	o									o	Close a file handle and save buffer to disk, if necessary
_CloseFileHandle_Y		o								o	Close a file handle and save buffer to disk, if necessary
_OpenFile	o			o	o		o			o	Open a file
_CreateFile	o			o	o		o			o	Create a file (returns error, if same file name exists on drive)
_CreateFile_CheckedNotExists	o			o	o		o			o	Create a file, but not existing same file name on drive must be tested before
_CreateFile_DeleteOld	o			o	o		o			o	Create a file and delete an existing file with same file name on drive
_DeleteFile				o						o	Delete an existing file on drive
_CheckFileInUse			o				o[*)]			o[**)]	Checks, whether a file (idendified by Dir Entry#) is just in use [*)] Z points to file handle, if file just in use [**)] C-Flag set, if file just in use
_CheckFileAccess_Mode		o			r25[1)]					o	Checks whether wanted operation[1)] suits to file open mode

8) Cluster

Routine & Label	Input-Setting / Output										
	File Handle	Cluster#	Sector#		Additional Input		Cluster#	Sector#		Error	
default Register	Y0/1	A			A		A			r25	
_GetNextCluster		A					A			o E1	Get next cluster# in a cluster chain of a file starting at cluster# in B
_GetNextCluster_Y	o						A			o E1	Get next cluster# in a cluster chain of a file starting at current cluster# in file handle
_GetFreeCluster					*)		A			o E2	Get a free cluster# from FAT *) Starting a cluster# in *gwLastSearchedCluster*
_getFreeCluster0									B		Count number of free clusters
_Sector2Cluster			A				A				Convert sector# to according cluster#
_Cluster2Sector		A						A			Convert cluster# to according # of first sector in cluster
_CheckUpdateFreeCluster		A							*)		Updates the variable *gwLastSearchedCluster* with A, if A is lower.

Error:
E1: cpNoNextCluster
E2: cpNoFreeCluster

9) Sector

Routine & Label	Input-Setting: Sector#	File Handle	Sector Pointer				Output: Sector Buffer	Sector#		Error	
default Register	A	Y0/1					Z	A		r25	
_GetSector	o						o			o	Load a sector to sector Buffer and save current sector if changed
_GetSector_File	o	o					o			o	Load a File sector to sector Buffer and save current sector if changed
_GetSector_FAT	o						o			o	Load a FAT sector to sector Buffer and save current sector if changed, if two FAT area exists, FAT-Copy is updated too
_GetSector_DIR	o						o			o	Load a Directory sector to sector Buffer and save current sector if changed
_GetNextSectorNumber	o							o	1)		Determine the next sector# to a current sector. If current sector is last sector of a cluster, following cluster is checked
_LoadNextFileSector		o								o	Load next sector of a file
_LoadNextFileSector_Position		o								o	Load next sector of a file and adjust file pointer to beginn of next sector
_AllocateNextFileSector		o						o		o	Allocate next sector for a file on drive
_AllocateFirstFileSector		o								o	Allocate first sector of a file (if file is empty)
_SaveFileSector		o								o	Save current file in buffer to the drive
_ClearFileSector		o									Clear current sector in file handle buffer
_SaveSector_FAT										o	Save current FAT Sector in buffer and FAT Copy if Number of FATs = 2
_SaveSector2FATCopy										o	Save current FAT Sector to FAT Copy if Number of FATs = 2
_SaveSector_DIR										o	Save current DIR Sector in buffer
_CheckAvailableSectorSize		o								o	Check available count of bytes available in current file sector R16-17: Count of Bytes; r24: file buffer status
_DriveReadSector_SEV	X					Z				o	Call Sector Read routine from Card-Driver and set DOS-Error Variable gbDOSError

9) Sector

Routine & Label	Input-Setting: Sector#	File Handle	Sector Pointer			Output: Sector Buffer	Sector#		Error	
default Register	A	Y0/1				Z	A		r25	
_DriveWriteSector_SEV	X		Z							Call Sector Write routine from Card-Driver and set DOS-Error Variable gbDOSError
_ReadDirSector										Call Sector Read Routine from Card-Driver to a load (Sector# in -> glDirSetorNumber) a Sector to Dir-Buffer

10) Directory

Routine & Label	Input-Setting: File Name	Dir Entry	Dir Entry#	Length of read/write	Additional Input	Output: Pointer for Output	Dir Entry	Dir Entry#	File Name	Error	
default Register	X	Z			A					r25	
_SearchDirEntry	o						o	o *)		o E1	Search Dir Entry# on Drive to a file name *) and stored in *gwDirEntry* too
_SearchNextDirEntry	o		*)					o **)		o E1	Search next Dir Entry# on Drive to a file name *) Starting after Dir Entry# stored in *gwDirEntry* (+1) **) and stored in *gwDirEntry* too
_GetFreeDirEntry			*)					o **)		o E2	Searching a free Dir Entry# on Drive *) Starting at Dir Entry# stored *gwFreeDirEntry* **) stored in *gwFreeDirEntry*
_GetDirEntry			*)				o				Get the Dir Entry (in Dir Sector) to a Dir Entry# *) to Dir Entry# in gwDirRootEntry
_ClearDirEntry		o									Clears a Dir entry in Dir Buffer
_CheckUpdateFreeDirEntry											Updates the variable *gwFreeDirEntry* with the lowest free Dir Entry
_InitDirEntry		o			1) 2)						Init Dir-Entry in Directory-Sector with file-name, attribut, first cluster#, date – and time-stamps 1) First-Cluster# 2) File-name in gsTempFileName
_NormFileName	X								*)	1)	Convert a file name to DOS-Convention: Upper-Case, 8+3, change Wildcard "*" to according number of "?" *) Result stored in *gsTempFileName*
_NormFileName0	X					Z				1)	Convert a file name to DOS-Convention: Upper-Case, 8+3, change Wildcard "*" to according number of "?"
_Comp_String_Mem	Z			r23*)	X**)					Z)	Compare a Dir-Entry string (Z) with a File-name string and consider Wildcard "?" *) maximum string-length **) String must end with string terminator Z) Z-Flag set if String matches (including wildcard "?")
_GetFileName		o				X+*)					Copy the File name from the Dir-Entry Buffer *) String variable with at least 8 bytes
_GetFileExtend		o				X+*)					Copy the File Extend from the Dir Entry Buffer *) String variable with at least 3 bytes

10) Directory

Routine & Label	Input-Setting: File Name	Dir Entry	Dir Entry#	Length of read/write	Additional Input	Output: Pointer for Output	Dir Entry	Dir Entry#	File Name	Error	
default Register	X	Z			A					r25	
_GetFileAttribut		o				X+*)					Copy the File Attribut from the Dir Entry Buffer *) Byte variable
_GetFileFirstCluster		o				X+*)					Copy the file first-cluster# from the Dir Buffer Entry *) Word Variable
_GetFileDate		o				X+*)					Get the File Date from the Dir Entry Buffer and convert it to Day, Month, Year *) 3 byte array with Day/Month/Year
_GetFileTime		o				X+*)					Get the File DTime from the Dir Entry Buffer and convert it to Second, Minute, Hour *) 3 byte array with Sec/Min/Hour
_GetFileSize		o				X+*)					Get the file size in Bytes from the Dir Buffer Entry *) Long Variable
_SetDirTimeDate		X*)									Write current Date and time to DIR-Entry *) X points to Time Entry
_GetNextDirSectorNumber			B					A			Get next Directory Sector#
_Check_Root											Checks whether current Directory is root directory Z-Flag set if true

1) Number of Wild-Card Letters in r25, Z-Flag set if r25=0 (no Wild-Card in file name)
Error:
E1: cpFileNotFound
E2: cpNoFreeDirEntry

11) Error- Dir-, FAT and file buffer status

Routine & Label	Input-Setting: Error Code	Hile handle	Input Variable				Output:	C-Flag	gbDOSError	Error	
default Register	R25	Y0/1								r25	
_SetDOSError	o							1	o		Write Error Code to AVR-DOS Variable gbDOSError and set C-Flag
_ClearDOSError								0	0		Write 0 to AVR-DOS Variable gbDOSError (no Error) and clear C-Flag
_SetErrorFlag								1			Set the AVR-DOS internal Error Flag (C-Flag)
SetErrorFlag_Add2toY								1			and adjust Y-Pointer by +2
_ClearErrorFlag								0			Clear the AVR-DOS internal Error Flag (C-Flag)
_ClearErrorFlag_Add2toY								0			and adjust Y-Pointer by +2
_SetFileNameError										*)	Set Error Code for “Wild-Card-Error in file name”
_SetBitFileStatusYZ		o	r24								Set Status-Bits in File Handle Status Byte
SetBitFileEmptyYZ		o									Set Bits for Status “Empty File” in File Handle Status Byte
SetBitEOFYZ		o									Set Bits for Status “End of File” in File Handle Status Byte
_ClearBitFileStatusYZ		o	R24								Clear Bits set in r24 in File Handle Buffer Status
_SetFileWritePending		o									Set Bit for Status “File Write Pending” in File Handle Status Byte
_SetDIRFATWritePending											Set Bit for Status “File DIR/FAT-Info Write Pending” in File Handle Status Byte
_CheckEOF		o						*)			Checks whether current file position pointer is in EOF position *) set C-Flag according to EOF and updates file buffer status

12) File handle variables

Routine & Label	Input-Setting: File Handle		Input Variable	Length of read / write			Output: Function Output	Set File Handle Pointer		
default Register	Y0/1				A					
_AddrFileHandleBase2Z	o							Z		
_AddrFileBuffer2X	o							X		
_AddrFileBuffer2Z	o							Z		
_AddrFileBufferStatus2Z	o							Z		
_AddrFileFirstCluster2X	o							X		
_AddrFilePosition2X	o							X		
_AddrFilePosition2Z	o							Z		
_AddrFileSectorNumber2X	o							X		
_AddrFileSectorNumber2Z	o							Z		
_AddrFileSize2X	o							X		
_AddrFileSize2Z	o							Z		
_LoadFileBufferStatusYZ	o			(1)			r24			
_LoadFileFirstClusterYZ2A	o			(4)			A			
_LoadFilePositionYZ2A	o			(4)			A			
_LoadFilePositionYZ2B	o			(4)			B			
_LoadFileSectorNumberYZ2A	o			(4)			A			
_LoadFileSizeYZ2A	o			(4)			A			

These Functions transfer variables of the file handle array to and from SRAM to the CPU-registers. Softstack pointed by Y+0/1 must contain the file handle base address.

_Addr…. Load Address of File Handle Variable to Pointer X or Z
_Load…. Load File Handle Variable from SRAM to Register
_Save…. Save (Store) File Handle Variable from register to SRAM

Kind of variable is included in routine-label behind the above mentioned prefixes

12) File handle variables

Routine & Label	Input-Setting: File Handle		Input-Setting: Input Variable	Input-Setting: Length of read / write			Output: Function Output	Output: Set File Handle Pointer		
default Register	Y0 / 1				A					
_LoadFileSizeYZ2B	o			(4)			B			
_SaveFileBufferStatusYZ	o		r24	(1)						
_SaveFileDirSectorNumberA2 2YZ	o		A	(4)						
_SaveFileFirstClusterA2YZ	o		A	(4)						
_SaveFilePositionA2YZ	o		A	(4)						
_SaveFilePositionB2YZ	o		B	(4)						
_SaveFileSectorNumberA2YZ	o		A	(4)						
_SaveFileSizeA2YZ	o		A	(4)						

13) Internal functions

Routine & Label	Input-Setting / Output										
	Input Variable	Input Variable	Input Variable	Length of read / write		Pointer for Output		Function Output		Error	
default Register											
_ClearFileSystem											Clears all SRAM used by AVR-DOS
_CheckFileSystemType	r24									o	Checks the MBR-file-type-system for valid to AVR-DOS C_ cleared if valid and T-Flag set to Fat16/32 (16:T=0, 32:T=1)
_CheckBootSign	r25									o	Checks the loaded sector in Directory-Buffer for the boot record sign (55AA at Byte 510+511)
_CheckA_Zero	A							ZF*)			Check A (R16-19) for Zero (ZF = Zero Flag)
_Comp16_r20_r22								ZF*) CF*)			Compare 2 Bytes (16 Bit) in r20/r21 with r22/23 (ZF=Zero Flag; CF=Carry Flag)
_CompareRange	Z	X	r24					ZF*) CF*)			Compares two Ranges in SRAM (Pointed by Z and X) with length r24 like cp (Z), (X)
_CompareRange_Update	Z	X	r24			X					Replace Range X with Z if Z < X
_ShiftR_WordB		B									
_ShiftL_WordB		B									
_LongX2A	X			(4)				A			Load 4 bytes (Long) from SRAM (Pointer in X) to A
_LongZ2A	Z			(4)				A			Load 4 bytes (Long) from SRAM (Pointer in Z) to A
_LongX2B	X			(4)				B			Load 4 bytes (Long) from SRAM (Pointer in X) to B
_LongZ2B	Z			(4)				B			Load 4 bytes (Long) from SRAM (Pointer in Z) to B
_A2LongX		A		(4)		X					Load 4 Bytes (Long) from A to SRAM (Pointer in X)
_A2LongZ		A		(4)		Z					Load 4 Bytes (Long) from A to SRAM (Pointer in Z)
_B2LongZ		B		(4)		Z					Load 4 Bytes (Long) from B to SRAM (Pointer in Z)
_WordZ2A	Z			(2)				A			Load 2 Bytes (Word) from SRAM (Pointer in Z) to A

13) Internal functions

Routine & Label	Input-Setting: Input Variable	Input Variable	Input Variable	Length of read / write		Output: Pointer for Output		Function Output		Error	
default Register											
_WordX2B	X			(2)				B			Load 2 Bytes (Word) from SRAM (Pointer in X) to B
_WordZ2B	Z			(2)				B			Load 2 Bytes (Word) from SRAM (Pointer in Z) to B
_A2WordX		A		(2)		X					Load 2 Bytes (Word) from A to SRAM (Pointer in X)
_B2WordX		B		(2)		X					Load 2 Bytes (Word) from B to SRAM (Pointer in X)
_B2WordZ		B		(2)		Z					Load 2 Bytes (Word) from B to SRAM (Pointer in Z)
_Mov4A2B		A		(4)				B			Copy 4 Bytes (Long) from A to B
_Mem_Copy	Z			r24		X					Copy SRAM (Pointer in Z) to SRAM (Pointer in X) with count of bytes in r24
_SaveA2LongY		A				Y					Copy A (4 Bytes) to Soft-Stack (decreases Y by 4)
_LongY2a	Y							A			Copy 4 Bytes from Soft-Stack to 4 (Y-Pointer unchanged)
_ClearA				(4)				A			Clear the registers A (r16-r19)
_Add32_Word		A	B					A			Add a Word value in B (r20-r21) to a Long Value in A (r16-r19)
_Add32_1		A						A			Add 1 to a Long value in A
_Add32_Byte		A	r20					A			Add a byte to a Long value in A
_Sub32_1		A						A			subtract 1 from a Long value in A