SAD Commands

command structure is Command [params] <"name"> {$Global options} {: additional definition} {| additional definition}

Commands	<H>	<H>	<H>	<H>																								
first 3 letters unique	address par1	address par2	address par3	address par4	Allowed item options for each comma																	max items	Symbol name	Global options				
					B	D	E	F	K	L	N	O	P	R	S	U	V	W	X	Y	=			A	C	Q	F	
args	start	end				x	x			x	x	x	x		x	x	x	x	x	x		32	n		x			
bank	bank	file offset	start addr	end addr																		0	n					
byte	start	end											x		x	x	x		x			1	y					
code	start	end																				0	n					
fill	start	end																				0	n					
function	start	end								x			x		x	x	x		x	x		2	n	x				
pswset	start	from																										
rbase	register	address	startr	endr																		0	n					
scan	start																					0	n					
structure	start	end				x	x			x	x	x	x	x	x	x	x	x	x	x		32	n	x		x		
subroutine	start					x	x			x	x	x	x		x	x	x	x	x	x	x	32	n		x			x
symbol	start	startr	endr		x			x										x				1	Y					
table	start	end										x	x		x	x	x	x		x		1	y	x				
text	start	end																				0	y					
timer (Note 1)	start	end																					y					
vector	start	end					x		x													1	n					
word	start	end											x		x	x	x		x			1	y					
xcode	start	end																				0	n					
setopts	<string>																					14	n					
clropts	<string>																					14	n					

Option letter definitions

Letter	Parameters			min	max
B	<D>	Bit		0	15
D	<H>	offset address			
E	<D> <R>	Encoded address, type, base register		1	4
F		Flags symbol		(Note 3)	
K	<D>	banK	0,1,8,9	(Note 4)	
L		Long			
N		Look for symbol Name			
O	<D>	repeat cOunt		2	31
P	<D>	Print field width		2	31
R		Reference (pointer) may need ‘K’ in multibanks			
S		Signed			
U		Unsigned (default)			
V	<F>	diVisor (floating point)			
W		Word			
W		Write Symbol (SYM command)			
X		Flip print – decimal to hex and back			
Y		bYte			
=	<R>	defines a subroutine answer for printout. (use with Y,W,S,U for size)			

<R> is register (in hex), <D> is decimal, <H> is hex

Notes

1 Timer structure command is under development, it may become a ‘struct’ perhaps with extra options

2 subroutines can have special functions, with signatures enabled, this detection is automatic. (see definition below)

3 Flags symbol sets the pseudo code display to always show the operand as separate bit fields, with names for all relevant opcodes

4 'K’ (Bank) is used where a pointer refers to a different bank than the current item address. This can happen with vector lists and pointers

Option letters for Global options

A		print in Args format (one item per line)	
C		print in Compact format (multiple items per line)	
F	<string>	Special Function (see right)	
Q	<n>	Terminator Bytes (1-3)	1 is default

command structure is Command [params] <"name"> {$Global options} {: additional definition} {| additional definition} {: additional definition} ...

Where - [ ] is a group of one or more hex values
{ } is zero or more groups of letters and numbers with defined start letter
A '$' char defines a single group of global options. Must be first.
':' and '|' chars define and delimit one data item, which may have multiple options, and occur multiple times
A '|' char causes a newline in the printout at the point, to allow layout options for long lists (=rows)

Command validation
<H> address must be valid for binary 0 - <max bank address>
<R> register must be valid for CPU 0- 0xFF for 8061, 0x3FF for 8065
<D> decimal values as defined in commands

Global option 'F' , special functions structure is :F < string> <pars>

Where -	string	params	size	meaning
	"uuyflu"	<R>	byte	unsigned in, unsigned out, function (1D) lookup. Function Address in register <R>
	"usyflu"	<R>	byte	unsigned in, signed out, function lookup Function Address in register <R>
	"suyflu"	<R>	byte	signed in, unsigned out, function lookup Function Address in register <R>
	"ssyflu"	<R>	byte	signed in, signed out, function lookup (1D) Function Address in register <R>
	"uuwflu"	<R>	word	unsigned in, unsigned out, function (1D) lookup. Function Address in register <R>
	"uswflu"	<R>	word	unsigned in, signed out, function lookup Function Address in register <R>
	"suwflu"	<R>	word	signed in, unsigned out, function lookup Function Address in register <R>
	"sswflu"	<R>	word	signed in, signed out, function lookup (1D) Function Address in register <R>
	"uytlu"	<R> <R>	byte	unsigned out, table lookup (2D) Function Address in first register, size in 2nd register
	"sytlu"	<R> <R>	byte	signed out, table lookup (2D)
	"uwtlu"	<R> <R>	word	unsigned out, table lookup (2D)
	"swtlu"	<R> <R>	word	signed out, table lookup (2D)

setopt command strings

string	default	meaning	
acomments	x	Auto comments	
default		the default combination of above (as ‘x’)	
funcnames	x	automatically name new functions (style is sub_xxxx)	
intrnames		automatically name interrupt handler subroutines	
labelnames		automatically add label names for all jumps	
manual		inhibit ALL automatic analysis, just use user commands	
sceprt	x	print pseudo source code	
signatures	x	look for signatures (special code sequences, eg, table lookup)	
ssubnames	x	automatically name new SPECIAL FUNCTION subroutines	
subnames	x	automatically name subroutines when called	
sympresets	x	Add preset symbol names for special registers	
tabnames	x	automatically name new tables	
8065		set SAD for 8065 CPU. (8061 is default)	
D8065		set SAD for later 8065 CPU with extra address mode	SAD 4.0.7

Notes.

Auto names will never overwrite names defined with a SYM command.
SAD has a set of predefined symbol names for special registers, 0-0x10 for 8061, 0-0x22 for 8065
SAD has a set of predefined symbol names for interrupt handler subroutines

SAD Commands

Commands	Information
args	define a list of arguments at specified address
bank	define a bank location within a .bin file
byte	define one or more byte values at specified address
code	define code at specified address
fill	define 'filler' (unused) at specified address
function	define a 'function' data structure at address (a 1 dimension lookup)
pswset	define where the PSW is set for a conditional jump (used to override SAD's autodetect – fixes "if (0=0)" style printouts)
rbase	define a register as a 'fixed base' lookup address. May be set for limited address range.
scan	specify an address to be [test] scanned as code
structure	define a data structure. Can be simple or complex.
subroutine	define a subroutine beginning at specified address
symbol	define a symbol at specified address, can be a bit, read or write, and may be set for limited address range
table	define a 'table' data structure at address (a 2 dimension lookup)
text	define text at specified address (typically a copyright message)
timer (Note 1)	define an extended timer structure. Under development.
vector	define a list of pointers to subroutines, typically defines a 'task list' of jobs
word	define one or more word values at specified address
xcode	define an area as NOT CODE. Useful sometimes to prevent data being interpreted as code.
setopts	set processing options
clropts	clear processing options

Comment commands (_cmt file)

<H> <text> where <H> is the opcode address, and <text> is the comment text added to the end of that line

e.g. 2037 # Watchdog Timer reset will result in a line like this -

2037: 11,05 clrb R5 WDG_Timer = 0; # Watchdog Timer reset

Comment text supports the use of special char sequences in the text as follows

\ indicates start of special sequence

\n insert a newline at this point
\w Wrap. Insert a newline, and then pad out to comments start column
\1 Print operand 1 in the comment – this will be printed same as in the opcode.
\2
\3
\s A Symbol. Has additional parameters
Full sequence is \s <H> for a symbol at address <H>
\s <H> : <D> for a bit symbol, bit <D> at address <H>
?? range and read or write?? Undecided as yet

\p A symbol with padding

\\ print a single '\' character