

Sheet1

To use enter the 3 character code for the ram location followed by a 2 character code for the value.  
 The value can range from 0 - 255 and is represented in converted hex.  
 To determine the value to use, convert the value to hex and then look up the letters in the conversion table.

Examples

Decimal	Hex	Code	Conversion Table	
10	0A	A0	0 A	8 M
73	49	FN	1 C	9 N
184	B8	PM	2 D	A O
			3 E	B P
			4 F	C R
			5 H	D S
			6 I	E T
			7 L	F U

RAMBASE	ADP	Start of usable RAM
ININTRO	ADP	1 byte TRUE in building intro, FALSE not - used to control NMI
VOICE1A	ADR	2 byte ROM address for next data byte for voice 1 - MSB NULL = Not Used
VOICE1C	ADT	1 byte counter for voice 1, when it hits zero get next data
VOICE2A	ADU	2 byte ROM address for next data byte for voice 2 - MSB NULL = Not Used
VOICE2C	AEC	1 byte counter for voice 2, when it hits zero get next data
VOICE3A	AED	2 byte ROM address for next data byte for voice 3 - MSB NULL = Not Used
VOICE3C	AEF	1 byte counter for voice 3, when it hits zero get next data
VOICE4A	AEH	2 byte ROM address for next data byte for voice 4 (NOICE) - MSB NULL = Not Used
VOICE4C	AEL	1 byte counter for voice 4, when it hits zero get next data
REPEAT1	AEM	2 Bytes to address of current repeat in Voice ? sound file
COUNT1	AEO	1 byte holding the count of the current repeat for voice ? sound file
REPEAT2	AEP	2 Bytes to address of current repeat in Voice ? sound file
COUNT2	AES	1 byte holding the count of the current repeat for voice ? sound file
REPEAT3	AET	2 Bytes to address of current repeat in Voice ? sound file
COUNT3	AFA	1 byte holding the count of the current repeat for voice ? sound file
REPEAT4	AFC	2 Bytes to address of current repeat in Voice ? sound file
COUNT4	AFE	1 byte holding the count of the current repeat for voice ? sound file

Sheet1

VOL1	AFF	1 byte holding the volume for voice 1
VOL2	AFH	1 byte holding the volume for voice 2
VOL3	AFI	1 byte holding the volume for voice 3
VOL4	AFL	1 byte holding the volume for voice 4
JIFFY	AFM	1 byte counter always counting down
SECONDS	AFN	1 byte counter counts down for seconds
INVINCE	AFO	1 byte holding Number of seconds left for invincibility
STICKJUMP	AFP	3 bytes Contains JP Joystick read
BADDIEJUMP	AFT	3 bytes Contains JP BDY.BLD.?
WAITING	AHC	1 byte holding how many jiffies with no movement
PAUSE	AHD	1 byte holding the current pause value for window closing
GAMEOVER	AHE	1 byte holding TRUE if game over
MOVEALONG	AHF	1 byte holding TRUE if we need to start the game
DROPSPEED	AHH	1 Byte holding how fast things fall
WINDOWSPEED	AHI	1 Byte holding how fast windows close
MUTE	AHL	1 byte holding if sounds are on / off
WARNCLOSING	AHM	1 byte true will warn when windows closing, false it will not
CONTROLLER	AHN	1 Byte holding SINGLE or DUAL
DUALSTATUS	AHO	1 byte holding the last DUAL controller movement
DUALPATTERN	AHP	2 bytes holding current DUAL controller pattern
SHOWHINT	AHS	1 byte TRUE to show int, FALSE to not - used by NMI
DEBOUNCE	AHT	1 byte TRUE if waiting for no stick movement
LEDGEBADDIES	AHU	1 Byte holding flag if we are doing Ledge baddies
LEDGECOUNTER	AIA	1 byte Counts down for moving baddies
LEDGEPOSITION	AIC	1 byte holding what row top ledge is on the screen
FLYINGCOUNTER	AID	1 byte Counts down for moving baddies
BOSSCOUNTER	AIE	1 byte counts down for boss movement
BOSSSIDE	AIF	1 byte What side boss is on LEFT, RIGHT or NONE
BOSSROW	AIH	1 byte holding the row the boss is on
BOSSCOL	AII	1 byte holding the boss column
BOSSTEXT	AIL	2 bytes hold address of text to display
BOSSLFONT	AIN	2 bytes Default left font address for boss
BOSSRFONT	AIP	2 bytes Default right font address for boss
BOSSLTEXT	AIS	2 bytes Default left text address for boss
BOSSRTEXT	AIU	2 bytes Default right text address for boss

Sheet1

STAGENUM	ALC	1 byte holding the stage (building or boss) in BLD.Defaults._stages
NEXTSTAGE	ALD	1 byte holding TRUE if moving to next stage after dieing (boss level)
BUILDINGNUM	ALE	2 bytes holding address to current building data
DIFFICULTY	ALH	1 byte difficulty
STARTFALLSPEED	ALI	1 byte hold the speed of falling 8, 16, 32, 64 etc
FALLSPEED	ALL	1 byte holding the current fallspeed counter
PAUSEVALUE	ALM	1 byte the PAUSE value for this difficulty
BONUSCOUNTER	ALN	2 bytes holding number of frames for bonuse counter change
FONTSCROLLDIR	ALP	1 byte Direction to scroll
FONTPOSITION	ALR	1 byte Current position in GAME.BuildingFont (0 - 15)
BLDFONT	ALS	2 bytes pointing at the font used for the current building
LIVES	ALU	1 byte number of lives
HEALTH	AMA	1 byte holding health 1 - 3
NEWLIFE	AMC	1 byte holding the 10,000's digit for a new life (5 or 0)
SCORE	AMD	8 bytes for the SCORE and a NULL
STEP	AMO	4 bytes for STEP and a NULL
BONUS	AMT	6 bytes for the BONUS and a NULL
CCLEVEL	ANF	Position of CC on the building 0 - 191 ( divide to get 0 - 23)
CCLEVELCHANGE	ANH	NULL for no change, else BALLOON or HELICOPTER
SCROLL	ANI	NULL if no scroll needed else UP or DOWN
SCROLLSTEP	ANL	What step we are in on the scroll (1 - 8) NULL means done
SCROLLCOUNTER	ANM	Count down for next scroll step
SLIDE	ANN	NULL if no sliding needed else LEFT or RIGHT
SLIDESTEP	ANO	What step we are in on the slide (1 - 8) NULL means done
SLIDECOUNTER	ANP	Count down for next scroll step
CHECK	ANR	If TRUE then main loop needs to check if CC is ok staying there
INVINCEDUR	ANS	1 byte holding how long invincibility lasts
BUFFER	ANT	16 byte buffer for various things
WINDOWCOLOR	AOT	1 byte holding the window color (used for hilighting pause)
WINDOWPAUSECLR	AOU	1 byte holding the color of a paused window
WINDOWCOUNTER	APA	1 byte Count down to window movement
WINDOWFLAG	APC	Holds the 9 bytes that show are flags for the windows
SPRITETABLE	APO	128 byte sprite table