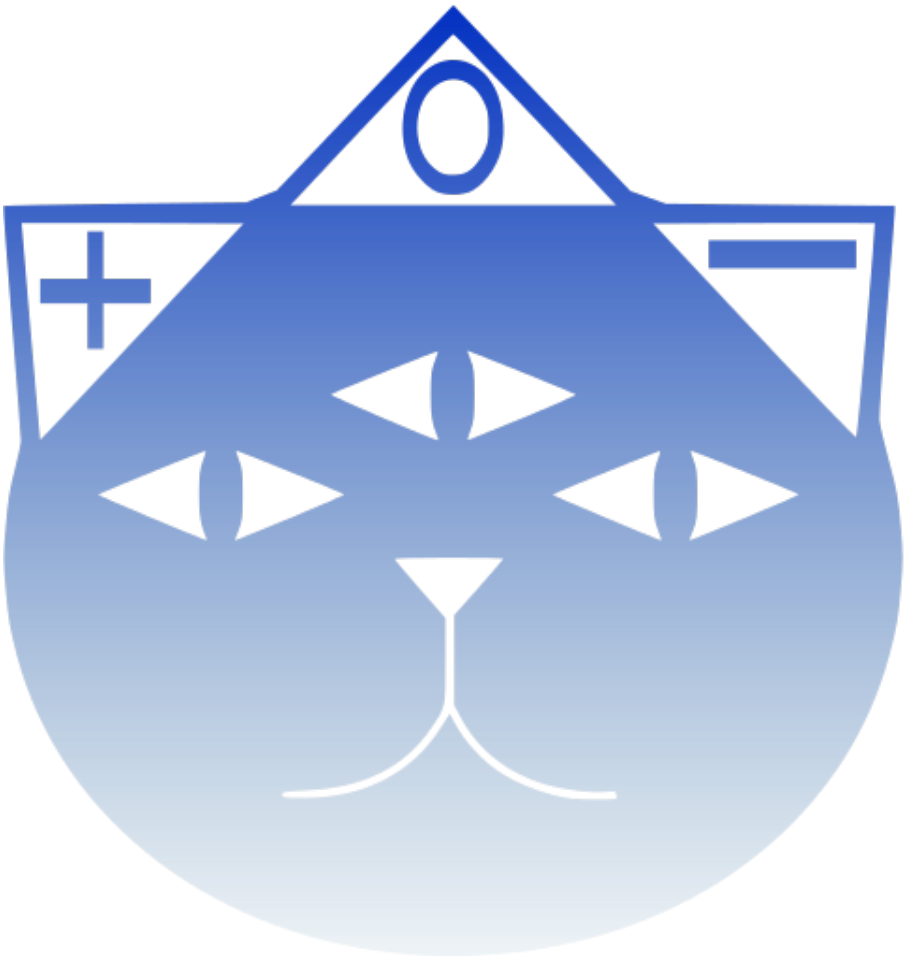


SBTCVM User Manual

Version: Mark 2-1.10

For SBTCVM Mark 2.0.3

Originally Written By:

Thomas Leathers

Table of Contents

History

version mark2-1.0: initial version

version mark2-1.1 add text encoding and command shell documentation.

version mark2-1.2 add SBTCVM Manifesto

version mark2-1.3 add Opcode list, other improvements.

version mark2-1.4 add glossary, other improvements

version mark2-1.5 document new memory pointer feature

version mark2-1.6 add new details on memory pointer feature.

version mark2-1.7 add section on configuration.

version mark2-1.8 Various Changes for the v2.0.2 version of SBTCVM.

version mark2-1.9 First round of changes for v2.0.3

version mark2-1.10 Various changes and updates.

SBTCVM Manifesto

Written By: Thomas Leathers
Late June, 2017
revision 1

Balanced Ternary is a strange base number. It traces far back in computer history, but not much has ever come of it other than a few footnotes. That is until recently, as several small projects such as multiplexers, appeared across the interwebs.

SBTCVM arose out of a curiosity for balanced ternary mathematics and computing, as did libbaltcalc. This reflects heavily in SBTCVM and its overall design, and goals. The learning curve of SBTCVM can be steep. In fact, programming SBTCVM has been a challenge in itself, due to not many resources or tools related to balanced ternary being available. SBTCVM hopes to change that by providing an evolving codebase and design to provide ideas as well as the software tools needed, to develop balanced ternary further. This evolving nature is evident in planned projects such as a portable balanced ternary programming language, and even an operating system for SBTCVM.

The sheer lack of tools and software support for balanced ternary can make itself quite evident. It therefore comes as no surprise to see people so passionate about the smallest balanced base number. It is hoped that SBTCVM can fill much of that void of support and tools. with a powerful integer mathematics backend, and a powerful command shell, and the VM itself, SBTCVM, continues to advance towards more powerful, features and tools.

The lack of advanced balanced ternary hardware isn't helping anything. Sure one could try using discrete components to create a balanced ternary computer. but to say that building 19,683 9-trit words of memory using discrete parts is a tad complicated, is an understatement. Hence SBTCVM's virtual machine. SBTCVM uses instruction level simulation for two main reasons. A lack of reference hardware and speed.

To conclude, SBTCVM is moving forward. New features, bugfixes, better documentation, and more. Perhaps some day, Balanced ternary computers will be available to the average curious user, but until then, SBTCVM will continue to advance ever further towards that overreaching goal.

Overview

How it Began

A word from Thomas Leathers, SBTCVM's first and lead developer.

SBTCVM started with a simple thought. Wondering what a balanced ternary computer would be like. Failing a search on the internet, I decided to use the integer mathematics library I had written prior to make my own VM. With Mark 2, SBTCVM is more capable than ever, and there are many features of SBTCVM that can get quite complex.

What Is SBTCVM?

SBTCVM is a balanced ternary virtual machine written in python, with graphics, sound and user input powered by pygame. It uses a fast Instruction Level simulation Method, has many tools and features, and has an extensive code base and backend.

SBTCVM Mark 2 Specifications:

- 6-trit instruction word.
- 9-trit data word
- 19,683 words of executable memory space.
- 9-trit IO bus.
- 729 9-trit words of scratch memory.
- 27x27 6-trit, 729 color RGB, plotter-like, display.
- 9x9 2-trit monochrome, plotter-like, display.
- 72 column by 54 line TTY.
- 3 voice polyphonic square wave sound generation.
- support for up to 6, bank switched sections of executable memory, 19,683 words each.
- The virtual CPU supports up to 9 active threads.

Getting Started

Launcher

SBTCVM's launcher, ***launcher.py***, acts as a central hub for SBTCVM's graphical utilities. Starting in v2.0.3, it brings you to the SBTCVM Desktop, a platform on which various smaller utilities are built, as SDAPs. See: SBTCVM Desktop

Settings

Settings, (settings.py) is a graphical configuration utility.

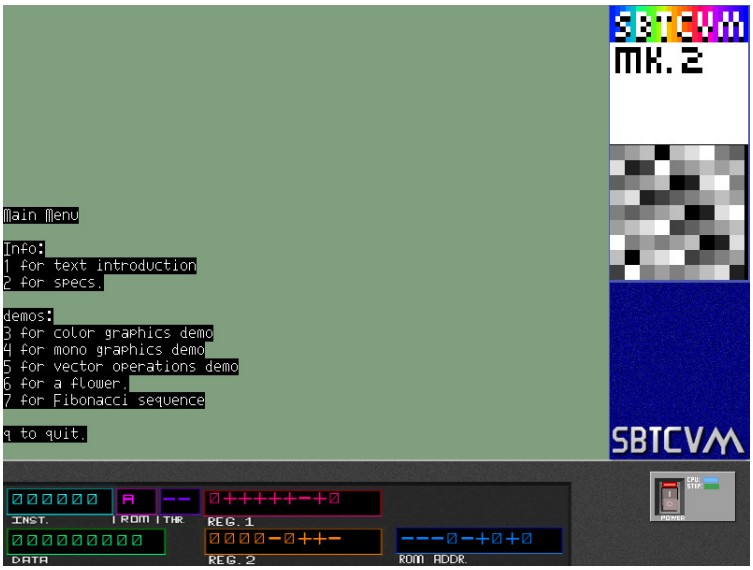
Fileview

Fileview is SBTCVM's integrated file browser. (***fileview.py***) it acts as a useful way to launch SBTCVM troms and stregs, as well as view tasm code, troms, stregs, logs, and even images using its companion viewers.

Introduction program

SBTCVM also features an introduction program that runs in the VM itself. You will find it in the “Get Started” menu.

You will see a TTY (the big text screen), and two small raster graphics displays to the right. Also, you will notice the readouts at the bottom. Feel free to check out The Introduction Program's selection of demos and information, and press ESCAPE to check out the pause menu.



Huds & Menus

VM User Interface

The Display

A:	TTY: This is the VM's text display. Its also mirrored to standard output (prefixed with "TTY|")
B:	27x27 pixel 6-trit RGB display
C:	9x9 pixel 2-trit monochrome display
D:	ROM indicator: (A-F) shows the current selected bank in exec. memory.
E:	current Data word
F:	current Instruction word
G:	CPU Register 1
H:	CPU Register 2

I:	Current Execution address.
J:	CPU status Indicator: Blue=active Orange=Standby
K:	Step-by-step debug mode indicator
L:	Active Thread ID

Keyboard controls

Key	description
F2	Toggle Step-by-step debug mode
F4	toggle Status Display
F7	dump raster displays
F8	screenshot
F10	manual memory dump
ESCAPE	Bring up pause menu.

VM Status & Other Messages

Sometimes, the VM itself will print to the TTY.
Here are some Status Messages and what they mean:

Status Message	description
VM SYSHALT: soft stop.	This SYSHALT message is triggered by the "STOP" instruction. usually a program will use this when it needs to shut down the VM
VM SYSHALT: User stop.	This SYSHALT is triggered by the user pressing the ESCAPE key and selecting "Stop VM"/"return to main menu"
REG1 DUMP:	The program has told the VM to dump the current state of CPU Register 1.
REG2 DUMP:	The program has told the VM to dump the current state of CPU Register 2.
VM SYSHALT: THREAD COLLISION!	The program attempted to start a thread ID that is already active.
VM SYSHALT: NO ACTIVE THREADS!	The program has killed and/or stopped all active threads.
VM SYSHALT: DIVIDE BY ZERO!	The program tried to divide by zero. Yes, even in SBTCVM: no dividing by zero :p
VM SYSHALT: T-ACT FAULT	an emulation bug has lead to the current thread being an inactive thread in an unexpected way. (aka you normally won't see this)
VM SYSHALT: MEMORY POINTER OVERFLOW	SBTCVM's memory pointer has exceeded the max positive integer (MPI) or MNI of 9 trits. This is most likely a fault of the running trom.

SBTCVM VM and Utilities

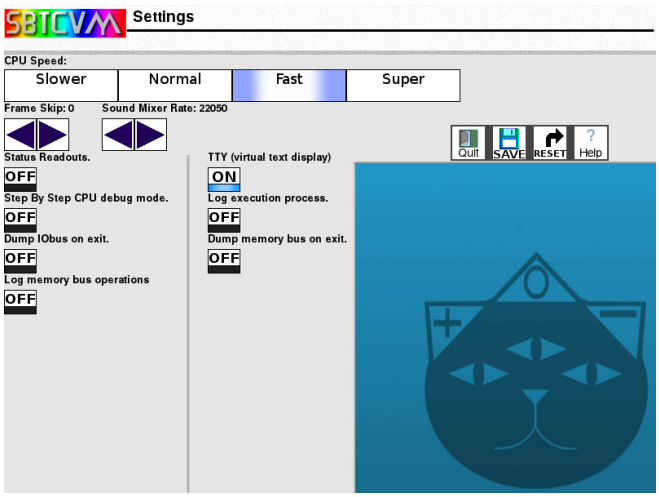
Program filename (program name) [command Shell command(if any)]	Description	command line options (if any)
Important Programs		
SBTCVM_MK2.py (SBTCVM Mark 2)	This is the core program of SBTCVM Mark 2. This is the	(this has no command line options. please see MK2-RUN.py)

	program thats launched by MK2-RUN and MK2-MENU to execute balanced ternary programs. Given that this is the Virtual machine itself, its a bit complex. That said, if stuck, or if the program that the VM is running, is not responding, press ESCAPE to bring up a nice pause menu.	
SBTCVM-asm2.py (SBTCVM's Assembler) [asm]	used to compile SBTCVM assembly sourcecode into TROMs SBTCVM-asm2.py will automatically search the subdirectories: "ROMS", "VMUSER", and "VMSYSTEM" if the tasm file is not found in SBTCVM's base directory. see SBTCVM assembly overview	SBTCVM-asm2.py -h (--help): this text SBTCVM-asm2.py -v (--version) SBTCVM-asm2.py -a (--about): about SBTCVM-asm2.py SBTCVM-asm2.py -c (--compile) [sourcefile]: build a tasm source into a trom SBTCVM-asm2.py -t (--tracecompile) [sourcefile]: same as -c but logs the compiling process in detail in the CAP directory. SBTCVM-asm2.py [sourcefile]: build a tasm source into a trom
SBTCVM VM and utility launchers		
settings.py	This is the normal way to configure SBTCVM to your preferences.	
launcher.py	This Is SBTCVM's main launcher. it acts as a central hub for SBTCVM	
fileview.py	This is SBTCVM's integrated file browser. useful for launching troms and viewing assembly code among other things.	
MK2-CS.py (SBTCVM Command Shell)	SBTCVM's custom command shell. This is by far the most powerful of SBTCVM's interfaces. SBTCVM's other command line tools, as well as the main menu, can be started from here. it also features balanced ternary mathematics commands that provide an interface to the underlying libbaltcalc library. currently the Command Shell is	This is MK2-CS.py, a command shell for SBTCVM Mark 2 commands: MK2-CS.py -h (--help) (help): this text MK2-CS.py -v (--version) MK2-CS.py -a (--about): about MK2-RUN.py

	designed to be run from the command line.	
MK2-TOOLS.py (SBTCVM graphical tool launcher.) [tools], [t]	Provides a command line interface to launch various SBTCVM graphical tools and is a means of testing things like the pause menu directly from the command line, or command shell.	This is MK2-TOOLS.py, a command line tools launcher for SBTCVM Mark 2 commands: MK2-RUN.py -h (--help) (help): this text MK2-RUN.py -v (--version) MK2-RUN.py -a (--about): about MK2-RUN.py MK2-RUN.py about : run menu about screen. MK2-RUN.py btclock : run Balanced Ternary clock. MK2-RUN.py pause : test pause menu
MK2-RUN.py (SBTCVM Command line launcher) [run]	launches the VM engine with the specified trom, or with the specified streg file. (auto-detected) SBTCVM will automatically search the subdirectories: "ROMS", "VMUSER", and "VMSYSTEM", and "ROMS" in "VMSYSTEM" if the file is not found in SBTCVM's base directory.	MK2-RUN.py -h (--help): this text MK2-RUN.py -v (--version) MK2-RUN.py -r (--run) [trom file]: run a trom as TROMA MK2-RUN.py -a (--about): about MK2-RUN.py MK2-RUN.py [trom file]: run a trom as TROMA MK2-RUN.py -r (--run) [streg file]: run specified streg file. MK2-RUN.py [streg file]: run specified streg file. MK2-RUN.py -le (--log_exec): same as -r but activates exec logging. MK2-RUN.py -tos: report aprox operations/second at end of execution.
Other Programs and utilities		
MK2-GFX.py (SBTCVM Graphics Toolkit Utility) [gfx]	Provides Binary raster image conversion.	MK2-GFX.py -h (--help): this text MK2-GFX.py -v (--version) MK2-GFX.py -a (--about): about MK2-GFX.py MK2-GFX.py -c (--colraster) [imagefile]: convert a 27x27 pixel or smaller image to color raster instrucions (exported as a *.tasm) MK2-GFX.py -cg (--colraster_groupcolor) [imagefile]: same as -c, but groups colors together as a compression scheme. MK2-GFX.py -cg2 (--colraster_groupcolor2) [imagefile]: same as -cg, but tracks the most common color and uses a single fill instruction to further compress the image.

MK2-GAT.py (SBTCVM Graphics Adapter Toolkit) [gat]	Provides image conversion tools. such as converting binary images to SBTGA Framebuffer data.	MK2-GAT.py -h (--help) (help): this text MK2-GAT.py -v (--version) MK2-GAT.py -a (--about): about MK2-GAT.py MK2-GAT.py -g0 (--modegfx0) [imagefile]: convert an (exactly) 114x81 pixel image to a full framebuffer area (exported as a *.tasm) for the G0, 114x81, 3-trit RGB screen mode. MK2-GAT.py -g2 (--modegfx2) [imagefile]: full framebuffer: G2 (114x81 1-trit monochrome) MK2-GAT.py -g3 (--modegfx3) [imagefile]: full framebuffer: G3 (54x38 3-trit RGB) MK2-GAT.py -g4 (--modegfx4) [imagefile]: full framebuffer: G4 (54x38 1-trit monochrome)

Configuring SBTCVM



SBTCVM features a graphical configuration utility, called **settings.py**.

Across the top are **CPU clock speed** presets. higher speeds are only faster if the host (the computer you are running SBTCVM on) has the cpu power.

next we have **Frame Skip** and the **sound mixer rate**. To change these, cycle through the available options using the arrow buttons.

The rest of the options just need the switch icon clicked. (it will either say ON or OFF).

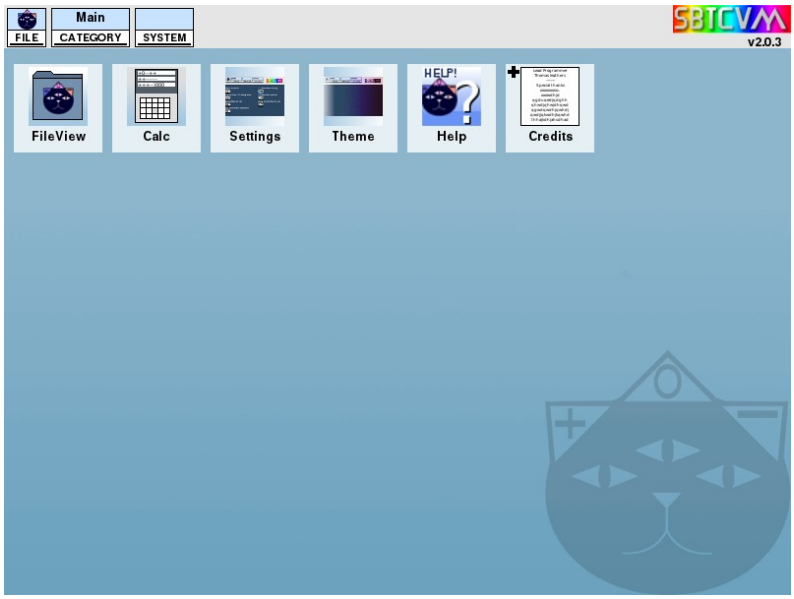
Status Readouts sets the default state of the status readouts area.

TTY controls wether the TTY is rendered. *(note that the TTY output can still be viewed if you run SBTCVM from a terminal).*

SBTCVM Desktop

Overview

The SBTCVM Desktop, otherwise known as launcher.py, acts as both a central hub for the VM and utilities, and as a powerful multitasking system.



Screenshot 1: SBTCVM Desktop with default settings.

Starting Programs

Starting an application is as simple as finding it in the selection of categories (use the category menu to navigate the various categories), and clicking it's tile. Depending on whether its an external utility like Helpview, or an SBTCVM desktop application like Console, It will either open in a new Host OS window, or a new SBTCVM Desktop window respectively.

Also, Some tiles point to TROMs and STREGs. These will launch in SBTCVMs Virtual Machine: SBTCVM Mark 2

Working with applications.

To move an application, click and drag it by its title bar. To close it, click the suspicious looking "X" button in the window title area. Its like most other bare-bones window managers, Its just in a pygame window.

When a window is active its title bar will be a different color from the rest. to change active windows, simply click anywhere in the desired window. Remember that clicking the "X" button will close it.

If you lost track of a window, Perhaps try the "Bring to top" option in taskman.

Built-ins:

Taskman.

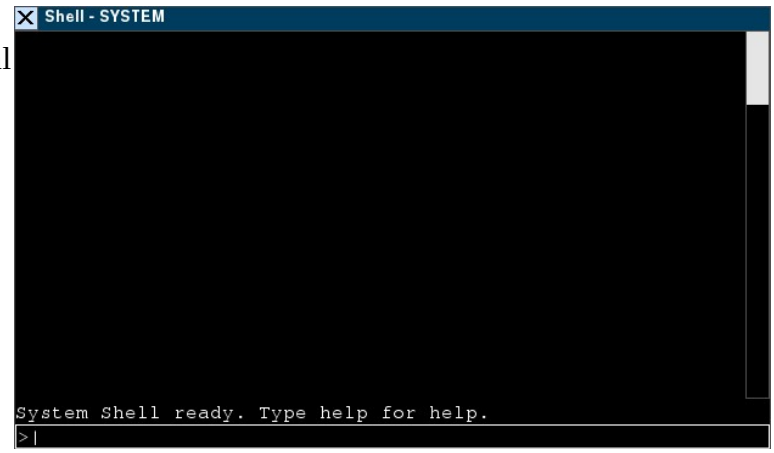
Taskman is well, a task manager. You can select a task by clicking the desired task in the list, and either Bring it to the top (and reset its x and y coordinates), or close it. *Tip: find it in the system menu.*

Console.

Console provides a convenient view of the Console. basically, its a system message board. *Tip: find it in the system menu.*

Shell.

Shell is the basis for the various Command Line Interfaces found on the SBTCVM Desktop. Its special que signals provide a quite simple means for creating CLI apps.



System Shell:

The System Shell is powered by the "Shell" built-in, and provides some system-related commands. *Tip: find it in the system menu.*

SDAPs

SBTCVM Desktop Application Plugins are essentially applications that "plug in" to the SBTCVM Desktop. Specifically, they are SBTCVM Desktop applications that consist of a plugin file, and an associated directory, and optionally more companion plugins.

Installing SDAPs is easy. just copy the associated directory and *.sdap.py files to the "plugins" directory, and restart the SBTCVM Desktop. They will load automatically. If the plugin doesn't seem to load, or if an error dialog appears when its running or being started, Its likely an issue with that plugin.

A warning:

As with any plugin system, SDAPs downloaded from untrusted sources may not be safe. So be sure to only download SDAPs from sources you can trust. Feel free to ask an SBTCVM developer if you need advice on where to find SDAPs.

The SBTCVM Project, its developers, contributors, users, testers, and researchers, ARE NOT responsible for ANY damages relating to or caused by, either directly or indirectly, untrusted SDAP plugins.

If you want to learn more about the terms and conditions that SBTCVM is distributed under, please see the README.md file included with SBTCVM, and the associated licenses.

Writing SDAPs

Basic Idea:

An SDAP contains 1 SDA, or SBTCVM Desktop Application, and some variables configuring it.

An SDA is basically a python class with a specific set of standardized methods. Each instance of an SDA therefore, is one python class instance.

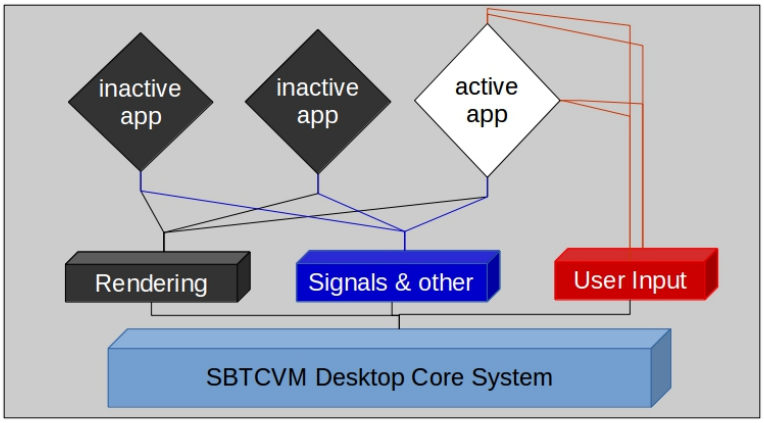


Illustration 1: Logic Diagram visualizing the 3 main processing areas of the desktop, and how they interact with the SDAs

Main Signaling:

render

The render method is called every core system cycle, making it crucial for real-time operations such as position calculations. Each instance is responsible for the frame rendering and rect generation. Don't worry, convenient functions for these are provided to do these automatically. Specifically the "getframes" and "drawframe" functions. You can check out SBTCVM's included SDAPs to see some real examples.

If you return an empty list, or a list of pygame rects, the system will treat the rects of that list as the areas to update, during passive rendering.

Note: Ensure the rects are oriented to the main screen surface, not the window surface.

Note: System will still do full updates on occasion, and not just on window moves and resizes either.

movet

The movet method is called whenever the core system needs to move that window by an abstract offset. The math is very simple, and can be found in SBTCVM's various included SDAPs.

SBTGA

Overview

SBTCVM Balanced Ternary Graphics Adapter

SBTGA is one of the latest additions to the VM's architecture.

Mode list

2 character ID	implemented?	description	color mode	ternary mode id
SB		default on system power-on. (TTY mode)	N/A	000000000
G0	Yes (v1)	114x81 framebuffered	3-trit RGB	---------
G1	No	114x81 framebuffered	3-trit monochrome	--------0
G2	Yes (v1)	114x81 framebuffered	1-trit monochrome	--------+
G3	Yes (v1)	54x38 framebuffered	3-trit RGB	-------0-
G4	Yes (v1)	54x38 framebuffered	1-trit monochrome	-------00

programming tips:

1: image conversion

conversion from binary image formats to full and partial SBTGA images:

(uses the GIMP image editor)

To achieve optimum results in the lower color depths offered by SBTGA v1, the extras/GIMP directory features a selection of several color palettes. for color modes of less than 256 you can use the interpolation feature in gimp to convert the image into indexed color using the provided GIMP palettes.

To convert images, see the Graphics Adapter Toolkit, or GAT: MK2-GAT.py

2: image swapping

It's FAR faster and simpler to move the display memory offset than it is to move images.

3: display updates

Updating the display from memory is not as simple as writing to the correct IO address under the hood. That said, try limiting screen updates to a reasonable interval. SBTCVM is a Virtual Machine after all.

Special Note: SBTGA will reference the current ram bank of the thread that calls its update address. so keep that in mind.

4: color modes

Sure, 3-trit RGB is plenty colorful, but it also eats up 3k words a framebuffer, if you have way too many graphics to show, or way too much code, 1-trit monochrome may not be colorful, but it does cut gfx memory usage to a third of what 3-trit RGB uses... so choose your color modes wisely...

Programming SBTCVM

SBTCVM assembly overview

Notice: This aspect of SBTCVM can get complex!!!
command line usage:
This is SBTCVM-asm2.py, SBTCVM Mark 2's assembler.
commands:
SBTCVM-asm2.py -h (--help): this text
SBTCVM-asm2.py -v (--version)
SBTCVM-asm2.py -a (--about): about SBTCVM-asm2.py
SBTCVM-asm2.py -c (--compile) [sourcefile]: build a tasm source into a trom
SBTCVM-asm2.py -t (--tracecompile) [sourcefile]: same as -c but logs the compiling process in detail in the CAP directory.
SBTCVM-asm2.py [sourcefile]: build a tasm source into a trom

for example, a source file called "file.tasm", will be compiled into the SBTCVM TROM: "file.trom"

it is possible to edit the trom directly, but can be more difficult.

tracelogs (enabled by compiling using the -t option) are based on the name of the source file and are located in the CAP directory.
example: intro.tasm would have a tracelog called intro-tasm-comp.log

instructions that need 6-trit RGB colors as data (listed below) can use this special formatting:
R,G,B where R G and B are respective color channels with values ranging from 0 to 255
example:
colorfill|255,255,255 is white

colorfill
setcolorreg
TTYbg

note: while SBTCVM-asm does make programming for SBTCVM mark 2 easier
a basic understanding of both how conventional computers work,
and balanced ternary mathematics, is still needed.

comments:
comments can be achieved as such:
"#comment"

Multiline text blocks:

start command:
"textstart"

end command:
"textstop"

to pad memory space, use "null".

SBTCVM-asm will fill any remaining TROM space with "null" automatically.

basic commands:

most of these correspond to SBTCVM Mk2 Opcodes:
any data specified should be specified as such:
'null|?????????'
SBTCVM has a pointer feature for data-based gotos and data/instruction modifiers.

-where ? is a balanced ternary digit
-Mark 2 uses 9-trit data. so does the assembler.
-note the vertical bar "|"
-also do note that a semicolon ";" is interchangeable with vertical bars in SBTCVM assembly as of assembler version 2.3.0
-note omitting the vertical bar and data will cause the data to default to ground. (000000000)
-for compatibility reasons, six trit integers are padded like so: 000??????

define a pointer like this:
'null|000000000|thisisanexample'
note: data can be omitted in situations it is unused. i.e.:
'null||thisisanexample'
In this case the data will be filled with ground (000000000)

on certain instructions you may point to a pointer as such:
'gotodata|>thisisanexample'
'gotodataif|>thisisanexample'
'YNgoto|>thisisanexample'
'setdata|>thisisanexample'
'setinst|>thisisanexample'
'romread1|>thisisanexample'
'romread2|>thisisanexample'

remember:
pointers differ from SBTCVM's memory pointer register. they are two completely different things.

similar shortcuts exist for various IOaddress points:

SBTCVM assembly IObus address keywords:

keyword(s)	description	read	write
random	random integer port	returns random 9-trit integer	N/A
mem1 - mem729	scratch memory	recover value	store value
dispmode	get/set SBTGA display mode register	N/A	set mode
dispoffset	get/set SBTGA DMA address register. (used to specify start of Video memory.)	N/A	set offset
dispupdate	SBTGA display update line.	N/A	instruct SBTGA to update the display using the configured range of memory, and the configured display mode.

SBTCVM assembly instruction list

instruction name	description:
"romread1"	------ ROMread 1 (first register)
"romread2"	-----0 ROMread 2 (second register)
"IOread1"	-----+ IOread 1 (first register)
"IOread2"	----0- IOread 2 (second register)
"IOwrite1"	----00 IOwrite 1 (first register)

"IOwrite2"	----0+ IOwrite 2 (second register)
"regswap"	----+- swap primary registers
"copy1to2"	----+0 copy register 1 to 2
"copy2to1"	----++ copy register 2 to 1
"invert1"	---0-- invert register 1
"invert2"	---0-0 invert register 2
"add"	---0-+ add
"subtract"	---00- subtract
"multiply"	---000 multiply
"divide"	---00+ integer division
"setreg1"	---0+- set register 1 -supports pointer referencing (see overview)
"setreg2"	---0+0 set register 2 -supports pointer referencing (see overview)
"setinst"	---0++ set instruction of rom address DATA using register 1 [IIIIII???] -supports pointer referencing (see overview)
"setdata"	---+-- set data of rom address DATA using register 1 [DDDDDDDDD] -supports pointer referencing (see overview)
"continue"	---+++ go to instruction following last goto instruction.
color raster display commands	
"colorpixel"	--0--- COLORDISP draw pixel using DRAWCOLOR using data ???XXXYYY ((Tip: draw same color pixels together between color changes)
"setcolorreg"	--0--0 SET DRAWCOLOR using data ???CCCCCC where C is a 6-trit RGB

	color :D
"colorfill"	--0--+ fill COLORDISP using data ???CCCCCC where C is a 6-trit RGB color :D
"setcolorvect"	--0-0- set colordisp vector register (for filled rectangle and line instructions) ???XXXYYY
"colorline"	--0-00 draw color line ???XXXYYY
"colorrect"	--0-0+ draw color filled rectangle ???XXXYYY
mono raster display commands	
"monopixel"	--0-+- MONODISP draw pixel using data ???XXYYMM where M=is a 2 trit monochrome value
"monofill"	--0-+0 fill COLORDISP using data ???????MM where M=is a 2 trit monochrome value
"setmonovect"	--0-++ set monodisp vector register (for filled rectangle and line instructions ???XXYY??
"monoline"	--00-- draw line ???XXYYMM where MM is a 2 trit greyscale color value.
"monorect"	--00-0 draw filled rectangle ???XXYYMM where MM is a 2 trit greyscale color value.
End of raster display commands	
"stop"	--000- stop (shuts down VM)
"null"	000000 null command
"gotodata"	--000+ goto data specified ROM address -supports pointer referencing (see overview)
"gotoreg1"	--00+- goto reg1 specified ROM address
"gotodataif"	--00+0 goto data specified ROM address IF REG 1 & 2 ARE EQUAL -supports pointer referencing (see overview)
"gotoifgreater"	--0+0-

	goto data specified ROM address IF REG 1 is greater than REG 2. -supports pointer referencing (see overview)
"wait"	--00++ wait from 0 to 19.682 seconds (exceeding this range causes an assemble error) usage: wait|:seconds note the colon. causes ONLY the current thread to wait, the threads will switch as normal.
"YNgoto"	--0+-- YN user goto. asks user to confirm goto. (ROM Address data specified) (best used with a label explaining what it does.) -supports pointer referencing (see overview)
"userwait"	--0+-0 user wait.
"TTYclear"	--0+-+ TTY clear
Exec Bank Switching Gotos:	
"gotoA"	--+--- A mandatory goto
"gotoAif"	--+--0 A reg equal goto
"gotoB"	--+--+ B mandatory goto
"gotoBif"	--+-0- B reg equal goto
"gotoC"	--+-00 C mandatory goto
"gotoCif"	--+-0+ C reg equal goto
"gotoD"	--+-+- D mandatory goto
"gotoDif"	--+-+0 D reg equal goto
"gotoE"	--+-++ E mandatory goto
"gotoEif"	--+0-- E reg equal goto
"gotoF"	--+0-0 F mandatory goto
"gotoFif"	--+0-+

	F reg equal goto
End Exec Bank Switching Gotos.	
"dumpreg1"	--++0+ dump register 1 to TTY
"dumpreg2"	--+++- dump register 2 to TTY
"TTYwrite"	--+++0 tty write port (direct) (???CCCCCC) where CCCCCC is the SBTCVMtext 6-trit character code. or you can use TTYwrite|:? where ?=single character (or "space" for a space or "enter" for a newline)
"buzzer"	--++++ buzzer port (direct) ???TFFFFF where T is time and FFFFF is frequency code time codes: + 0.3 seconds 0 0.2 seconds - 0.1 seconds
"setregset"	-0-000 (see regset system)
"regset"	-0-00+ (see regset system)
"TTYlinedraw"	toggles The regset flag controlling wether the tty updates on each character, or just on newlines. (SBTCVM Mark 2 defaults this flag to per character) use TTYlinedraw|on to enable and TTYlinedraw|off to disable.
"TTYbg"	set TTY background color through regset. SET using data ???CCCCCC where C is a 6-trit RGB color :D (6-trit color enhanced)
"TTYmode"	set TTY line mode through regset use "TTYmode|54" for 54 line mode (9x9 font) (Mark 2 uses this mode by default) , use "TTYmode|27" for 27 line mode. (9x18 font) both of these modes are 72 columns and both run at 648x486
"setkeyint"	-0-+++ (see keyscan system) set keyinterrupt register. (used to select a key interrupt to activate.) ?????DDDD=4-trit scancode

	you can use setkeyint|:? where ?=codelabel (see keyscan system)
"keyint"	-00--- (see keyscan system) keyboard interrupt. (data)DDDDDDDDD=rom address to goto on interrupt.
"offsetlen"	-0-++0 A is either "on" or "off" if on, the offset length system will use ground states in place of the states of the destination data. (default is off.) leave off if you are doing things like manipulating part of a number, B is 0-8 and controls offset from radix point (default 0) C is 1-9 and controls the offset length trit length. (default is 9) offsetlen|A,B,C see offset length for more information.
"clearkeyint"	-00--0 (see keyscan system) clear key interrupt| ????????0=clear keyinterrupt chosen by ketinterrupt, ????????+=clear all keyinterrupts.
Threading Instructions	see Threading for more information.
"threadref"	--+00- set threading system refrence register TT or ???????TT
"threadstart"	--+000 launch a thread at address AAAAAAAAA. use threadref to specify what thread. can use SBTCVM-asm goto references to specify thread starting address.
"threadstop"	--+00+ stop current thread. when no threads are active the VM will halt.
"threadkill"	--+0+- stop thread referenced by threadref register
Memory Pointer Operations	
ptread	read from address in memory pointer to reg1
ptwri	write reg1 data to address in memory pointer
ptwridat	write current data word data to address in memory pointer --supports pointer referencing (see overview) (doesn't affect address of write, affects data written)
ptinc	increment memory pointer by 1
ptdec	decrement memory pointer by 1

ptset	set memory pointer with reg1
ptadd	add reg1 to memory pointer
ptget	copy memory pointer to reg1
3 Voice Sound "chip"	(uses square waves) Music tip: the wait instruction can be a huge help.
sfreq	-00-++ sfreq|v:freq where "freq" is a decimal value (in Hz) between 1 and 2187 DO NOT label it with Hz... that will cause an error. where "v" is either 1, 2, or 3. for each of the three voices. sets the given frequency on the given voice.
splay	-00-++ splay|v where "v" is either 1, 2, or 3. for each of the three voices. plays at freq previously chosen for that voice., (defaults to 1094 Hz at VM start-up)
sstop	-00-++ sstop|v where "v" is either 1, 2, or 3. for each of the three voices. stops the specified voice.

Memory Pointer

SBTCVM mark 2 features a memory pointer feature. you can use this to iterate through large sets of data for example.

operations:

operation	tasm instruction	opcode
-read to reg 1 using memory pointer	ptread	-0-0+0
write to memory location using memory pointer and reg1	ptwri	-0-0++
-write to memory location using memory pointer and data portion of word.	ptwridat	-0-+--
-memory pointer control.	see below	-0-0+-

memory pointer control instruction control modes:

mode	control code	tasm instruction
-mode to add a +	--???????	ptinc
-mode to add a -	-0???????	ptdec
-mode to set with reg1	-+???????	ptset
-mode to add reg1 to pointer	0-???????	ptadd
-mode to copy pointer to reg1	00???????	ptget

note: exceeding either The MPI or MNI (+++++++++ or ---------, respectively) of 9 trits will cause a "MEMORY POINTER OVERFLOW!" VM SYSHALT condition.

Libbaltcalc API

Functions:

BTTODEC(btinteger)	Convert Balanced ternary integer to decimal
DECTOBT(decinteger)	Convert Decimal integer to balanced ternary
btmul(numA, numB)	Multiply two balanced ternary integers
btadd(numA, numB)	Add two balanced ternary integers
btsub(numA, numB)	Subtract two balanced ternary integers
btdivcpu(numA, numB)	Divide two balanced ternary integers. Returns “ZDIV” upon zero division.
btdiv(numA, numB)	Divide two balanced ternary integers. Returns “Zero Division Error” upon zero division.
BTINVERT(numtoinvert)	Invert balanced ternary integer to its opposite. i.e. “-+” would invert to “+-”
progbiasand(polarset, inpA, inpB)	A "programmable" biased and gate. returns a positive if: input a (inpA) = input b (inpB) = polarity line (polarset) else it returns zero
polarityand(inpA, inpB)	A polarized and gate returns + if both input A (inpA) and input B (inpB) = + returns - if both input A (inpA) and input B (inpB) = - otherwise it returns zero
progbiasor(polarset, inpA, inpB)	A programmable biased or gate returns "+" if either or both inputs equal the polarity line (polarset) else it returns "0"
progbiasnor(polarset, inpA, inpB)	A programmable biased orn gate returns "+" if either equal the polarity line (polarset) returns "0" either if neither or both inputs equal the polarity line (polarset)

btint class:

Syntax:

int1=btint("+-")

int2=btint(2)

int3=btint(int2)

the btint class provides support for python's builtin integer mathematics methods, and some others methods

invert() (returns the balanced ternary inversion. i.e. -+0- becomes +-0+)

dec() (explicitly returns a normal python integer)

bt() (explicitly returns balanced ternary integer in string form.)

supported python methods and notes about implementation of them.

__str__ (returns balanced ternary integer in string form.)

__int__ (returns a normal python integer)

(these mathematics methods return btint instances.)

__add__

__sub__

__truediv__, __div__, __floordiv__ (limited to integers only as floating point operations are not yet supported)

__mul__

__abs__

__neg__

__pos__

__invert__ (returns the balanced ternary inversion. i.e. -+0- becomes +-0+)

Offset Length Control

opcode:"-0-++0"

data usage:
????ABBCC

tasm: offsetlen|off,0,9

set A to "0" to base output of load and set operations on current data of destination.
set A to "+" to base output of load and set operations on a ground state.

B is a 2-trit balanced ternary integer, and sets the offset from the radix of load and set operations.
(default is -- (0))

--=0
-0=1
-+=2
0-=3
00=4
0+=5
+-=6
+0=7
++=8

A is also a 2-trit balanced ternary integer, and controls the truncation length of load and set operations.
(default is ++ (9 trits))

--=1
-0=2
-+=3
0-=4
00=5
0+=6
+-=7
+0=8
++=9

Keyscan System

-0-+++|set keyinterrupt register. (used to select a key interrupt to activate.)
 ?????DDDD=4-trit scancode

-00---|keyboard interrupt. (data)DDDDDDDDD=rom address to goto on interrupt.

-00--0|clear key interrupt| ????????0=clear keyinterrupt chosen by ketinterrupt,
 ????????+=clear all keyinterrupts.

related SBTCVM Assembly instructions

instruction	
"setkeyint"	-0-+++ set keyinterrupt register. (used to select a key interrupt to activate.) ?????DDDD=4-trit scancode you can use setkeyint|:? where ?=codelabel
"keyint"	-00--- keyboard interrupt. (data)DDDDDDDDD=rom address to goto on interrupt.
"clearkeyint"	-00--0 clear key interrupt| ????????0=clear keyinterrupt chosen by keyinterrupt, ????????+=clear all keyinterrupts.

4-trits are used, and keyboard is only alphanumeric (with exception to return,
+, -, and space) to keep VM less complex. an extended mode should be at least
6-trits, and preferably be a separate mode.

SBTCVM Mark 2 Keyboard interrupt codes:

key	4-trit keycode
1	----
2	---0
3	---+
4	--0-
5	--00

6	--0+
7	--+-
8	--+0
9	--++
0	-0--
-	-0-0
+	-0-+
a	-00-
b	-000
c	-00+
d	-0+-
e	-0+0
f	-0++
g	-+--
h	-+-0
i	-+-+
j	-+0-
k	-+00
l	-+0+
m	-++-
n	-++0
o	-+++
p	0---
q	0--0
r	0--+
s	0-0-
t	0-00
u	0-0+
v	0-+-
w	0-+0
x	0-++
y	00--
z	00-0
space	00-+
enter	000-

Regset System

instruction	instruction description	assembly instruction
-0-000	set regset pointer register	"setregset"
-0-00+	regset operation (DDDDDDDDD) (data)	"regset"

additional documentation of regset system:

any SBTCVM SYSTEM registers deemed not important enough to have an exclusive opcode should be added to the regset system.

step 1: set regset pointer register (-0-000) (tasm: "setregset") using current data
step 2: set selected register with regset operation (-0-00+) (tasm: "regset") using current data

register pointers:

pointer	name	description	assembly shortcut instruction	assembly notes:
---------	TTY BG color	(???CCCCCC) (6-trit RGB) DEFAULT=000000 set the background of the TTY	"TTYbg"	6-trit color enhanced
--------0	TTY render mode	????????0=normal ???????? +=draw-on-newline	"TTYlinedraw"	use "TTYlinedraw|on" to enable draw-on-newline, use TTYlinedraw|off to enable normal mode.
--------+	TTY line mode	????????0=54 line mode (default) ????????+=27 line mode	"TTYmode"	use "TTYmode|54" for 54 line mode (9x9 font) (Mark 2 uses this mode by default) , use "TTYmode|27" for 27 line mode. (9x18 font) both of these modes are 72 columns and both run at 648x486

NOTES ABOUT USE IN SBTCVM ASSEMBLY

Threading

Introduction:

SBTCVM programs can run up to 9 independent threads at once.

by default, all operations are run in one thread, called the "main thread"
the other 8 threads are called "background threads"

the main thread takes the ID "--" note that each thread has a 2-trit thread ID.
Trying to open a thread with the same ID as another active thread will raise a VM SYSHALT Thread Collision exception.

registers and other data sets and values that are thread-unique:

regset pointer
CPU register 1
CPU register 2
Execute Address
goto continue address
thread reference register
color Vector register
mono vector register
color display color value register
offset length controll settings.

SBTCVM assembly instructions:

Threading Instructions	
"threadref"	--+00- set threading system refrence register TT or ???????TT
"threadstart"	--+000 launch a thread at address AAAAAAAAA. use threadref to specify what thread. can use SBTCVM-asm goto references to specify thread starting address.
"threadstop"	--+00+ stop current thread. when no threads are active the VM will halt.
"threadkill"	--+0+- stop thread referenced by threadref register

notes:

- Only the main thread can use the builtin Keyboard interrupts, so keep that in mind.
- SBTCVM switches through each thread in order, and switches threads each clock cycle.
- Remember that the more threads are running, the slower SBTCVM runs overall.

STREG files

What is STREG?

streg files allow multi-trom programs to be launched with much more control.

what does it look like?

Example (excerpt from intro.streg)

```
#SBTCVM trom execution group file
#this is appended to the SBTCVM window title.
streg_subtitle="Introduction to SBTCVM"
TROMA="intro.trom"
#TROMB="DEFAULT.TROM"
#TROMC="DEFAULT.TROM"
#TROMD="DEFAULT.TROM"
#TROME="DEFAULT.TROM"
#TROMF="DEFAULT.TROM"
```

notice that even though `"intro.trom"` is in the `"VMSYSTEM"` directory, its still loaded.

troms stored in SBTCVM's base directory, the `"ROMS"` directory, and the `"VMSYSTEM"` directory can be loaded using just the base filename. e.g. `"intro.trom"`

SBTCVM Command Shell

The command shell (MK2-CS.py) is the powerful command shell of the SBTCVM project.

Overview of commands:

run	pass-through to MK2-RUN.py runs sbtcvm troms and stregs. use “run help” for more help.
asm	pass-through to SBTCVM-asm2.py assembles tasm source files into troms. use “asm help” for more help.
gfx	pass-through to MK2-GFX.py various graphics conversion functions use “gfx help” for more help
tools (t)	SBTCVM tools test suite. (MK2-TOOLS.py) use “t help” for more help
mainmenu	start SBTCVM's main menu
list [type]	The main uses of list are to list important file types like troms, SBTCVM assembly files, and so on, that SBTCVM can see without explicit paths. list types : list known important file type keywords list paths : list paths searched by sbtcvm for important file types list [type] : look for the specified type.
help	Command shell help
version	version information
about	about information
quit	quit SBTCVM Command Shell

Mathematics Commands:

add	add two balanced ternary integers (separated by a space)
sub	subtract two balanced ternary integers (separated by a space)
div	divide two balanced ternary integers (separated by a space) (floored division only)
mul	multiply two balanced ternary integers (separated by a space)
btdec	convert a balanced ternary integer to decimal

decbt	convert a decimal integer to balanced ternary
invert	invert a balanced ternary integer
mpi	calculate the Max Positive Integer for a given number of trits
mni	calculate the Max Negative Integer for a given number of trits
mcv	calculate the Max Combinations Value for a given number of trits

SBTCVM Mark 2 IObus

RES= Reserved

RFI= Reserved For Implementation.

Scratch memory

Start:

`---------` [1] <-9841> assembly IO reference keyword: mem1

End:

`---++++++` [729] <-9113> assembly IO reference keyword: mem729

random

random 9-trit integer port (updated per clock cycle)

`--0------` [730] <-9112> assembly IO reference keyword: random

use this to get pseudo-random integers.

SBTGA IO Points.

--0---+-+ [750]

[mode]

--0---+0- [751]

[offset]

--0---+00 [752]

[update]

--0---+0+ [753]

[RES]

--0---++- [754]

[RES]

--0---++0 [755]

[RES]

--0---+++ [756]

[RES]

--0--0--- [757]

[RES]

--0--0--0 [758]

[RES]

--0--0--+ [759]

[RES]

--0--0-0- [760]

[RES]

--0--0-00 [761]

[RES]

--0--0-0+ [762]

[sprite 1] {RFI}

--0--0-+- [763]

[sprite 2] {RFI}

--0--0-+0 [764]

[sprite 3] {RFI}

SBTCVM-BTT-6 Text Encoding Specification

Information:

SBTCVM's text encoding is one of its older features. So its 6 trits. that means 729 single-tryte codes also future plans are in place for multi-tryte codes.

Its technical name in the SBTCVM project is:
SBTCVM-BTT-6
this Stands for:

- Simple
- Balanced
- Ternary
- Computer
- Virtual
- Machine
- -
- Balanced
- Ternary
- Text
- -
- 6 trits

notes:

- codes `++++--` to `++++++` are reserved for multi-tryte codes.
- single-tryte code 000000 is reserved as a "NULL" character

the encoding itself is defined in libSBTCVM.
This is the reference documentation for that encoding.

List of single-tryte Codes

```
------ a
-----0 b
```

```
-----+ c
----0- d
----00 e
----0+ f
----+- g
----+0 h
----++ i
---0-- j
---0-0 k
---0-+ l
---00- m
---000 n
---00+ o
---0+- p
---0+0 q
---0++ r
---+-- s
---+-0 t
---+-+ u
---+0- v
---+00 w
---+0+ x
---++- y
---++0 z
---+++ A
--0--- B
--0--0 C
--0--+ D
--0-0- E
--0-00 F
--0-0+ G
--0-+- H
--0-+0 I
--0-++ J
--00-- K
--00-0 L
--00-+ M
--000- N
--0000 O
--000+ P
```

```
--00+- Q
--00+0 R
--00++ S
--0+-- T
--0+-0 U
--0+-+ V
--0+0- W
--0+00 X
--0+0+ Y
--0++- Z
--0++0 0
--0+++ 1
--+--- 2
--+--0 3
--+--+ 4
--+-0- 5
--+-00 6
--+-0+ 7
--+-+- 8
--+-+0 9
--+-++ `
--+0-- ~
--+0-0 !
--+0-+ @
--+00- #
--+000 $
--+00+ %
--+0+- ^
--+0+0 &
--+0++ *
--++-- (
--++-0 )
--++-+ -
--++0- =
--++00 _
--++0+ +
--+++- [
--+++0 ]
--++++ \
-0---- {
```

```
-0---0 }
-0---+ |
-0--0- ;
-0--00 '
-0--0+ ,
-0--+- .
-0--+0 /
-0--++ :
-0-0-- "
-0-0-0 <
-0-0-+ >
-0-00- ?
-0-000 (NEWLINE)
-0-00+ (SPACE)
-0-0+-
-0-0+0
-0-0++
-0-+--
-0-+-0
-0-+-+
-0-+0-
-0-+00
-0-+0+
-0-++-
-0-++0
-0-+++
-00---
-00--0
-00--+
-00-0-
-00-00
-00-0+
-00-+-
-00-+0
-00-++
-000--
-000-0
-000-+
-0000-
-00000
```

```
-0000+
-000+-
-000+0
-000++
-00+--
-00+-0
-00+-+
-00+0-
-00+00
-00+0+
-00++-
-00++0
-00+++
-0+---
-0+--0
-0+--+
-0+-0-
-0+-00
-0+-0+
-0+-+-
-0+-+0
-0+-++
-0+0--
-0+0-0
-0+0-+
-0+00-
-0+000
-0+00+
-0+0+-
-0+0+0
-0+0++
-0++--
-0++-0
-0++-+
-0++0-
-0++00
-0++0+
-0+++-
-0+++0
-0++++
```

```
-+----
-+---0
-+---+
-+--0-
-+--00
-+--0+
-+--+-
-+--+0
-+--++
-+-0--
-+-0-0
-+-0-+
-+-00-
-+-000
-+-00+
-+-0+-
-+-0+0
-+-0++
-+-+--
-+-+-0
-+-+-+
-+-+0-
-+-+00
-+-+0+
-+-++-
-+-++0
-+-+++
-+0---
-+0--0
-+0--+
-+0-0-
-+0-00
-+0-0+
-+0-+-
-+0-+0
-+0-++
-+00--
-+00-0
-+00-+
-+000-
```

```
-+0000
-+000+
-+00+-
-+00+0
-+00++
-+0+--
-+0+-0
-+0+-+
-+0+0-
-+0+00
-+0+0+
-+0++-
-+0++0
-+0+++
-++---
-++--0
-++--+
-++-0-
-++-00
-++-0+
-++-+-
-++-+0
-++-++
-++0--
-++0-0
-++0-+
-++00-
-++000
-++00+
-++0+-
-++0+0
-++0++
-+++--
-+++-0
-+++-+
-+++0-
-+++00
-+++0+
-++++-
-++++0
```

```
-+++++
0-----
0----0
0----+
0---0-
0---00
0---0+
0---+-
0---+0
0---++
0--0--
0--0-0
0--0-+
0--00-
0--000
0--00+
0--0+-
0--0+0
0--0++
0--+--
0--+-0
0--+-+
0--+0-
0--+00
0--+0+
0--++-
0--++0
0--+++
0-0---
0-0--0
0-0--+
0-0-0-
0-0-00
0-0-0+
0-0-+-
0-0-+0
0-0-++
0-00--
0-00-0
0-00-+
```

```
0-000-
0-0000
0-000+
0-00+-
0-00+0
0-00++
0-0+--
0-0+-0
0-0+-+
0-0+0-
0-0+00
0-0+0+
0-0++-
0-0++0
0-0+++
0-+---
0-+--0
0-+--+
0-+-0-
0-+-00
0-+-0+
0-+-+-
0-+-+0
0-+-++
0-+0--
0-+0-0
0-+0-+
0-+00-
0-+000
0-+00+
0-+0+-
0-+0+0
0-+0++
0-++--
0-++-0
0-++-+
0-++0-
0-++00
0-++0+
0-+++-
```

```
0-+++0
0-++++
00----
00---0
00---+
00--0-
00--00
00--0+
00--+-
00--+0
00--++
00-0--
00-0-0
00-0-+
00-00-
00-000
00-00+
00-0+-
00-0+0
00-0++
00-+--
00-+-0
00-+-+
00-+0-
00-+00
00-+0+
00-++-
00-++0
00-+++
000---
000--0
000--+
000-0-
000-00
000-0+
000-+-
000-+0
000-++
0000--
0000-0
```

```
0000-+
00000-
000000 (null)
00000+
0000+-
0000+0
0000++
000+--
000+-0
000+-+
000+0-
000+00
000+0+
000++-
000++0
000+++
00+---
00+--0
00+--+
00+-0-
00+-00
00+-0+
00+-+-
00+-+0
00+-++
00+0--
00+0-0
00+0-+
00+00-
00+000
00+00+
00+0+-
00+0+0
00+0++
00++--
00++-0
00++-+
00++0-
00++00
00++0+
```

```
00+++-
00+++0
00++++
0+----
0+---0
0+---+
0+--0-
0+--00
0+--0+
0+--+-
0+--+0
0+--++
0+-0--
0+-0-0
0+-0-+
0+-00-
0+-000
0+-00+
0+-0+-
0+-0+0
0+-0++
0+-+--
0+-+-0
0+-+-+
0+-+0-
0+-+00
0+-+0+
0+-++-
0+-++0
0+-+++
0+0---
0+0--0
0+0--+
0+0-0-
0+0-00
0+0-0+
0+0-+-
0+0-+0
0+0-++
0+00--
```

```
0+00-0
0+00-+
0+000-
0+0000
0+000+
0+00+-
0+00+0
0+00++
0+0+--
0+0+-0
0+0+-+
0+0+0-
0+0+00
0+0+0+
0+0++-
0+0++0
0+0+++
0++---
0++--0
0++-+
0++-0-
0++-00
0++-0+
0++-+-
0++-+0
0++-++
0++0--
0++0-0
0++0-+
0++00-
0++000
0++00+
0++0+-
0++0+0
0++0++
0+++--
0+++-0
0+++-+
0+++0-
0+++00
```

```
0+++0+
0++++-
0++++0
0+++++
+-----
+----0
+----+
+---0-
+---00
+---0+
+---+-
+---+0
+---++
+--0--
+--0-0
+--0-+
+--00-
+--000
+--00+
+--0+-
+--0+0
+--0++
+--+--
+--+-0
+--+-+
+--+0-
+--+00
+--+0+
+--++-
+--++0
+--+++
+-0---
+-0--0
+-0--+
+-0-0-
+-0-00
+-0-0+
+-0-+-
+-0-+0
+-0-++
```

```
+-00--
+-00-0
+-00-+
+-000-
+-0000
+-000+
+-00+-
+-00+0
+-00++
+-0+--
+-0+-0
+-0+-+
+-0+0-
+-0+00
+-0+0+
+-0++-
+-0++0
+-0+++
+-+---
+-+--0
+-+--+
+-+-0-
+-+-00
+-+-0+
+-+-+-
+-+-+0
+-+-++
+-+0--
+-+0-0
+-+0-+
+-+00-
+-+000
+-+00+
+-+0+-
+-+0+0
+-+0++
+-++--
+-++-0
+-++-+
+-++0-
```

```
+-++00
+-++0+
+-+++-
+-+++0
+-++++
+0----
+0---0
+0---+
+0--0-
+0--00
+0--0+
+0--+-
+0--+0
+0--++
+0-0--
+0-0-0
+0-0-+
+0-00-
+0-000
+0-00+
+0-0+-
+0-0+0
+0-0++
+0-+--
+0-+-0
+0-+-+
+0-+0-
+0-+00
+0-+0+
+0-++-
+0-++0
+0-+++
+00---
+00--0
+00--+
+00-0-
+00-00
+00-0+
+00-+-
+00-+0
```

```
+00-++
+000--
+000-0
+000-+
+0000-
+00000
+0000+
+000+-
+000+0
+000++
+00+--
+00+-0
+00+-+
+00+0-
+00+00
+00+0+
+00++-
+00++0
+00+++
+0+---
+0+--0
+0+--+
+0+-0-
+0+-00
+0+-0+
+0+-+-
+0+-+0
+0+-++
+0+0--
+0+0-0
+0+0-+
+0+00-
+0+000
+0+00+
+0+0+-
+0+0+0
+0+0++
+0++--
+0++-0
+0++-+
```

```
+0++0-
+0++00
+0++0+
+0+++-
+0+++0
+0++++
++----
++---0
++---+
++--0-
++--00
++--0+
++--+-
++--+0
++--++
++-0--
++-0-0
++-0-+
++-00-
++-000
++-00+
++-0+-
++-0+0
++-0++
++-+--
++-+-0
++-+-+
++-+0-
++-+00
++-+0+
++-++-
++-++0
++-+++
++0---
++0--0
++0--+
++0-0-
++0-00
++0-0+
++0-+-
```

```
++0-+0
++0-++
++00--
++00-0
++00-+
++000-
++0000
++000+
++00+-
++00+0
++00++
++0+--
++0+-0
++0+-+
++0+0-
++0+00
++0+0+
++0++-
++0++0
++0+++
+++---
+++--0
+++--+
+++-0-
+++-00
+++-0+
+++-+-
+++-+0
+++-++
+++0--
+++0-0
+++0-+
+++00-
+++000
+++00+
+++0+-
+++0+0
+++0++
++++-- (reserved for multi-tryte codes)
++++-0 (reserved for multi-tryte codes)
```

```
++++-+ (reserved for multi-tryte codes)
++++0- (reserved for multi-tryte codes)
++++00 (reserved for multi-tryte codes)
++++0+ (reserved for multi-tryte codes)
+++++- (reserved for multi-tryte codes)
+++++0 (reserved for multi-tryte codes)
++++++ (reserved for multi-tryte codes)
```

SBTCVM Mark 2 opcode list

```
------|ROMread 1(first register)#some instructions from ------ to --+
+++ are legacy
-----0|ROMread 2(second register)
-----+|IOread  1(first register)
----0-|IOread  2(second register)
----00|IOwrite 1(first register)
----0+|IOwrite 2(second register)
----+-|swap primary registers
----+0|copy register 1 to 2
----++|copy register 2 to 1
---0--|invert register 1
---0-0|invert register 2
---0-+|add
---00-|subtract
---000|multiply
---00+|rounded devide
---0+-|set register 1
---0+0|set register 2
---0++|set instruction [dataspace is rom address] (using register 1
as instruction (IIII??)
---+--|set data [dataspace is rom address] (using register 1 as data
(DDDDDD)
```

```
---+-0

---+-+

---+0-

---+00

---+0+

---++-

---++0

---+++|continue execution at next address after previous goto.

--0---|COLORDISP draw pixel using DRAWCOLOR using data XXXYYY ((Tip:
draw same color pixels together between color changes)

--0--0|SET DRAWCOLOR using data CCCCCC where C is a 6-trit RGB
color :D

--0--+|fill COLORDISP using data CCCCCC where C is a 6-trit RGB color
:D

--0-0-|set colordisp vector register (for filled rectangle and line
instructions) XXXYYY

--0-00|draw line XXXYYY

--0-0+|draw filled rectangle XXXYYY

--0-+-|MONODISP draw pixel dusing data XXYYMM where M=is a 2 trit
monochrome value

--0-+0|fill COLORDISP using data ????MM where M=is a 2 trit
monochrome value

--0-++|set monodisp vector register (for filled rectangle and line
instructions XXYY??

--00--|draw line  XXYYMM where MM is a 2 trit greyscale color value.

--00-0|draw filled rectangle XXYYMM where MM is a 2 trit greyscale
color value.

--00-+

--000-|stop (shuts down VM)

--0000|legacy null

--000+|goto data specified ROM adress

--00+-|goto reg1 specified ROM adress

--00+0|goto data specified ROM adress IF REG 1 & 2 ARE EQUAL

--00++|wait a specified time from 0 to 19.682 seconds (encoded with
```

bias (++++++++++=19.682 seconds)

--0+--|YN user goto. asks user to confirm goto. (best used with a label)

--0+-0|user wait.

--0+-+|TTY clear

--0+0-|goto data specified ROM adress ID REG 1 > REG 2

--0+00

--0+0+

--0++-

--0++0

--0+++

--+---|A mandatory goto (the following gotos switch TROMS! make sure config setup is done properly!

--+--0|A reg equal goto

--+--+|B mandatory goto

--+-0-|B reg equal goto

--+-00|C mandatory goto

--+-0+|C reg equal goto

--+-+-|D mandatory goto

--+-+0|D reg equal goto

--+-++|E mandatory goto

--+0--|E reg equal goto

--+0-0|F mandatory goto

--+0-+|F reg equal goto

--+00-|threadref ???????TT where TT is the thread IS

--+000|start thread refrenced in threadref, at address AAAAAAAAA

--+00+|thread STOP: stop current thread

--+0+-|thread kill: kill thread refrenced in threadref.

--+0+0

--+0++

--++--

--++-0

```
--++-+
--++0-
--++00|
--++0+|dump register 1 to TTY
--+++-|dump register 2 to TTY
--+++0|tty write port (direct)
--++++|buzzer port (direct)
-0----|set copyrange start addr [reserved for implimentation]
-0---0|set copyrange end addr [reserved for implimentation]
-0---+|set copyrange dest addr [reserved for implimentation]
-0--0-|Exec > IO copy [reserved for implimentation]
-0--00|Exec > Exec copy [reserved for implimentation]
-0--0+|IO > Exec copy [reserved for implimentation]
-0--+-|IO > IO [reserved for implimentation]
-0--+0
-0--++
-0-0--
-0-0-0
-0-0-+
-0-00-
-0-000|regset pointer (see regset.txt)
-0-00+|regset operation (DDDDDDDDD) (data) (see regset.txt)
-0-0+-|memory point control
-0-0+0|read to reg 1 using memory pointer
-0-0++|write to memory location using memory pointer and reg1
-0-+--|write to memory location using memory pointer and data portion
of word.
-0-+-0|reserved - mempoint
-0-+-+|reserved - mempoint
-0-+0-
-0-+00
-0-+0+
```

-0-++-

-0-++0|readwrite offset length. control. (see offsetlength.txt)

-0-+++|set keyinterrupt register. (used to select a key interrupt to activate.) ?????DDDD=4-trit scancode (see keyboard-inter.txt)

-00---|keyboard interrupt. (data)DDDDDDDDD=rom address to goto on interrupt. (see keyboard-inter.txt)

-00--0|clear key interrupt| ????????0=clear keyinterrupt chosen by ketinterrupt, ????????+=clear all keyinterrupts.

-00--+

-00-0-

-00-00

-00-0+

-00-+-

-00-+0

-00-++|3 voice square synth, VCFFFFFFF V=voice code, C=command code (+=frequency, 0=play -=stop) F=linear encode from 1 to 2187 Hz

-000--

-000-0

-000-+

-0000-

-00000

-0000+

-000+-

-000+0

-000++

-00+--

-00+-0

-00+-+

-00+0-

-00+00

-00+0+

-00++-

-00++0

```
-00+++
-0+---
-0+--0
-0+--+
-0+-0-
-0+-00
-0+-0+
-0+-+-
-0+-+0
-0+-++
-0+0--
-0+0-0
-0+0-+
-0+00-
-0+000
-0+00+
-0+0+-
-0+0+0
-0+0++
-0++--
-0++-0
-0++-+
-0++0-
-0++00
-0++0+
-0+++-
-0+++0
-0++++
-+----
-+---0
-+---+
```

```
-+--0-
-+--00
-+--0+
-+--+-
-+--+0
-+--++
-+-0--
-+-0-0
-+-0-+
-+-00-
-+-000
-+-00+
-+-0+-
-+-0+0
-+-0++
-+-+--
-+-+-0
-+-+-+
-+-+0-
-+-+00
-+-+0+
-+-++-
-+-++0
-+-+++
-+0---
-+0--0
-+0--+
-+0-0-
-+0-00
-+0-0+
-+0-+-
```

```
-+0-+0
-+0-++
-+00--
-+00-0
-+00-+
-+000-
-+0000
-+000+
-+00+-
-+00+0
-+00++
-+0+--
-+0+-0
-+0+-+
-+0+0-
-+0+00
-+0+0+
-+0++-
-+0++0
-+0+++
-++---
-++--0
-++--+
-++-0-
-++-00
-++-0+
-++-+-
-++-+0
-++-++
-++0--
-++0-0
```

```
-++0-+
-++00-
-++000
-++00+
-++0+-
-++0+0
-++0++
-+++--
-+++-0
-+++-+
-+++0-
-+++00
-+++0+
-++++-
-++++0
-+++++
0-----
0----0
0----+
0---0-
0---00
0---0+
0---+-
0---+0
0---++
0--0--
0--0-0
0--0-+
0--00-
0--000
0--00+
```

```
0--0+-
0--0+0
0--0++
0--+--
0--+-0
0--+-+
0--+0-
0--+00
0--+0+
0--++-
0--++0
0--+++
0-0---
0-0--0
0-0--+
0-0-0-
0-0-00
0-0-0+
0-0-+-
0-0-+0
0-0-++
0-00--
0-00-0
0-00-+
0-000-
0-0000
0-000+
0-00+-
0-00+0
0-00++
0-0+--
```

```
0-0+-0
0-0+-+
0-0+0-
0-0+00
0-0+0+
0-0++-
0-0++0
0-0+++
0-+---
0-+--0
0-+--+
0-+-0-
0-+-00
0-+-0+
0-+-+-
0-+-+0
0-+-++
0-+0--
0-+0-0
0-+0-+
0-+00-
0-+000
0-+00+
0-+0+-
0-+0+0
0-+0++
0-++--
0-++-0
0-++-+
0-++0-
0-++00
```

```
0-++0+
0-+++-
0-+++0
0-++++
00----
00---0
00---+
00--0-
00--00
00--0+
00--+-
00--+0
00--++
00-0--
00-0-0
00-0-+
00-00-
00-000
00-00+
00-0+-
00-0+0
00-0++
00-+--
00-+-0
00-+-+
00-+0-
00-+00
00-+0+
00-++-
00-++0
00-+++
```

```
000---
000--0
000--+
000-0-
000-00
000-0+
000-+-
000-+0
000-++
0000--
0000-0
0000-+
00000-
000000|null
00000+
0000+-
0000+0
0000++
000+--
000+-0
000+-+
000+0-
000+00
000+0+
000++-
000++0
000+++
00+---
00+--0
00+--+
00+-0-
```

```
00+-00
00+-0+
00+-+-
00+-+0
00+-++
00+0--
00+0-0
00+0-+
00+00-
00+000
00+00+
00+0+-
00+0+0
00+0++
00++--
00++-0
00++-+
00++0-
00++00
00++0+
00+++-
00+++0
00++++
0+----
0+---0
0+---+
0+--0-
0+--00
0+--0+
0+--+-
0+--+0
```

```
0+--++
0+-0--
0+-0-0
0+-0-+
0+-00-
0+-000
0+-00+
0+-0+-
0+-0+0
0+-0++
0+-+--
0+-+-0
0+-+-+
0+-+0-
0+-+00
0+-+0+
0+-++-
0+-++0
0+-+++
0+0---
0+0--0
0+0--+
0+0-0-
0+0-00
0+0-0+
0+0-+-
0+0-+0
0+0-++
0+00--
0+00-0
0+00-+
```

```
0+000-
0+0000
0+000+
0+00+-
0+00+0
0+00++
0+0+--
0+0+-0
0+0+-+
0+0+0-
0+0+00
0+0+0+
0+0++-
0+0++0
0+0+++
0++---
0++--0
0++--+
0++-0-
0++-00
0++-0+
0++-+-
0++-+0
0++-++
0++0--
0++0-0
0++0-+
0++00-
0++000
0++00+
0++0+-
```

```
0++0+0
0++0++
0+++--
0+++-0
0+++-+
0+++0-
0+++00
0+++0+
0++++-
0++++0
0+++++
+-----
+----0
+----+
+---0-
+---00
+---0+
+---+-
+---+0
+---++
+--0--
+--0-0
+--0-+
+--00-
+--000
+--00+
+--0+-
+--0+0
+--0++
+--+--
+--+-0
```

```
+--+-+
+--+0-
+--+00
+--+0+
+--++-
+--++0
+--+++
+-0---
+-0--0
+-0--+
+-0-0-
+-0-00
+-0-0+
+-0-+-
+-0-+0
+-0-++
+-00--
+-00-0
+-00-+
+-000-
+-0000
+-000+
+-00+-
+-00+0
+-00++
+-0+--
+-0+-0
+-0+-+
+-0+0-
+-0+00
+-0+0+
```

```
+-0++-
+-0++0
+-0+++
+-+---
+-+--0
+-+--+
+-+-0-
+-+-00
+-+-0+
+-+-+-
+-+-+0
+-+-++
+-+0--
+-+0-0
+-+0-+
+-+00-
+-+000
+-+00+
+-+0+-
+-+0+0
+-+0++
+-++--
+-++-0
+-++-+
+-++0-
+-++00
+-++0+
+-+++-
+-+++0
+-++++
+0----
```

```
+0---0
+0---+
+0--0-
+0--00
+0--0+
+0--+-
+0--+0
+0--++
+0-0--
+0-0-0
+0-0-+
+0-00-
+0-000
+0-00+
+0-0+-
+0-0+0
+0-0++
+0-+--
+0-+-0
+0-+-+
+0-+0-
+0-+00
+0-+0+
+0-++-
+0-++0
+0-+++
+00---
+00--0
+00--+
+00-0-
+00-00
```

```
+00-0+
+00-+-
+00-+0
+00-++
+000--
+000-0
+000-+
+0000-
+00000
+0000+
+000+-
+000+0
+000++
+00+--
+00+-0
+00+-+
+00+0-
+00+00
+00+0+
+00++-
+00++0
+00+++
+0+---
+0+--0
+0+--+
+0+-0-
+0+-00
+0+-0+
+0+-+-
+0+-+0
+0+-++
```

```
+0+0--
+0+0-0
+0+0-+
+0+00-
+0+000
+0+00+
+0+0+-
+0+0+0
+0+0++
+0++--
+0++-0
+0++-+
+0++0-
+0++00
+0++0+
+0+++-
+0+++0
+0++++
++----
++---0
++---+
++--0-
++--00
++--0+
++--+-
++--+0
++--++
++-0--
++-0-0
++-0-+
++-00-
```

```
++-000
++-00+
++-0+-
++-0+0
++-0++
++-+--
++-+-0
++-+-+
++-+0-
++-+00
++-+0+
++-++-
++-++0
++-+++
++0---
++0--0
++0--+
++0-0-
++0-00
++0-0+
++0-+-
++0-+0
++0-++
++00--
++00-0
++00-+
++000-
++0000
++000+
++00+-
++00+0
```

```
++00++
++0+--
++0+-0
++0+-+
++0+0-
++0+00
++0+0+
++0++-
++0++0
++0+++
+++---
+++--0
+++--+
+++-0-
+++-00
+++-0+
+++-+-
+++-+0
+++-++
+++0--
+++0-0
+++0-+
+++00-
+++000
+++00+
+++0+-
+++0+0
+++0++
++++--
++++-0
++++-+
```

```
++++0-
++++00
++++0+
+++++-
+++++0
++++++
```

Glossary

#:

1-trit monochrome: n. A balanced ternary monochrome color mode using 1 trit. creating 3 shades.

2-trit monochrome: n. A balanced ternary monochrome color mode using 2 trits. creating 27 shades.

3-trit RGB: n. a balanced ternary color mode using 1 trit per color channel, creating 27 colors.

6-trit RGB: n. a balanced ternary color mode using 2 trits per color channel, creating 729 colors.

A:

B:

Balanced Ternary: n. the smallest balanced base number. A base number consisting of a positive digit, zero, and a negative digit. the base number SBTCVM uses.

C:

Command Shell: SBTCVM Mark 2's custom command shell interface that provides various utilities and mathematics operations.

D:

E:

F:

Fileview: n. SBTCVM's integrated file browser.

G:

goto reference: n. an SBTCVM assembly feature that allows automatic address calculation. use not limited to gotos. See assembly documentation for more information.

H:

I:

IObus: n. SBTCVM's Input Output Bus. Not to be confused with the Memory Bus.

Inversion: v. The process of replacing positive values with negative values and negative values with positive values, in a balanced ternary number. thus, Inverting the balanced ternary number. i.e. +0-- would become -0++.

IObus refrence: n. an SBTCVM assembly feature that provides, goto-reference-like IObus keywords. See assembly documentation for more information.

J:

K:

KiloTryte: n. 1093 Trytes. (KT)
Kilotrit: n. 1093 trits. (Kt)

L:

launcher: n. SBTCVM's Central launcher utility, *launcher.py*

M:

MPI: n. Short for Max positive integer. Refers to the maximum positive integer a length of trits can store.
MNI: n. Short for Max negative integer. Refers to the maximum negative integer a length of trits can store.
MCV: n. Short for Max combinations value, Refers to the maximum number of combinations of a length of trits. in balanced ternary, this differs from the MPI greatly.
Megatrit: n. 1093 Kilotrits. (Mt)
MegaTryte: n. 1093 KiloTrytes. (MT)
Mark 1: n. The original prototype series. used 4-trit instructions and 6-trit data. This series is no longer in development.
Mark 2: n. The mark 2 series features many new features and improvements. it uses 6-trit instructions and 9-trit data. Is in active development.
Mark 3: n. The in-planning Future Series. To feature 12-trit data. instruction width will be at least 6-trits. also planned is virtual disk support and an operating system and bios.
Memory Bus: n. SBTCVM's main memory bus. not to be confused with the IObus.

N:

O:

Offset Length: n. The name of SBTCVM's trit width, offset from radix control used to manipulate data read/write operations.

P:

Polarity Notation: n. The form of balanced ternary notation used by SBTCVM, using a "+" for the positive digit, a "0" for ground, and a "-" for the negative digit.

Q:

R:

regset: n. A general register system where less-important, registers are accessed.

S:

Settings: n. SBTCVM's configuration utility.

Scratch memory: a small portion of the IObus with scratch memory for general use. can store 729 9-trit words.

SBTCVM: n. The acronym for "Simple Balanced Ternary Computer Virtual Machine" the full name of SBTCVM.

SBTC-OS: n. The name of a planned future Operating system for the in-planning SBTCVM mark 3 series.

SBTCVM-BTT-6: SBTCVM's balanced ternary text encoding system.

T:

TDA: n. Ternary Demo Architecture. The temporary name of the Very first prototype of what became SBTCVM.

trit: n. A balanced ternary digit. Analogous to a binary bit. Can be either positive, ground, or negative. (t)
tryte: n. a set of 6 trits. Analogous to a Byte (T)
tasm: n. [1] The name of an SBTCVM assembly source file. [2] a shorter term for SBTCVM assembly.
trom: n. an SBTCVM rom image. in mark 2, these are loaded into ram and executed.

U:

V:

VM SYSHALT: A general message SBTCVM prints when some within-vm program condition, expected or not, halts the VM.

W:

X:

Y:

Z:

GNU Free Documentation License

GNU Free Documentation License
Version 1.3, 3 November 2008

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall

directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a

specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated

location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
D. Preserve all the copyright notices of the Document.
E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
H. Include an unaltered copy of this License.
I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements

   and/or dedications given therein.
L. Preserve all the Invariant Sections of the Document,
   unaltered in their text and in their titles. Section numbers
   or the equivalent are not considered part of the section titles.
M. Delete any section Entitled "Endorsements". Such a section
   may not be included in the Modified Version.
N. Do not retitle any existing section to be Entitled "Endorsements"
   or to conflict in title with any Invariant Section.
O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or
appendices that qualify as Secondary Sections and contain no material
copied from the Document, you may at your option designate some or all
of these sections as invariant. To do this, add their titles to the
list of Invariant Sections in the Modified Version's license notice.
These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains
nothing but endorsements of your Modified Version by various
parties--for example, statements of peer review or that the text has
been approved by an organization as the authoritative definition of a
standard.

You may add a passage of up to five words as a Front-Cover Text, and a
passage of up to 25 words as a Back-Cover Text, to the end of the list
of Cover Texts in the Modified Version. Only one passage of
Front-Cover Text and one of Back-Cover Text may be added by (or
through arrangements made by) any one entity. If the Document already
includes a cover text for the same cover, previously added by you or
by arrangement made by the same entity you are acting on behalf of,
you may not add another; but you may replace the old one, on explicit
permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License
give permission to use their names for publicity for or to assert or
imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this
License, under the terms defined in section 4 above for modified
versions, provided that you include in the combination all of the
Invariant Sections of all of the original documents, unmodified, and
list them all as Invariant Sections of your combined work in its
license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and
multiple identical Invariant Sections may be replaced with a single
copy. If there are multiple Invariant Sections with the same name but
different contents, make the title of each such section unique by
adding at the end of it, in parentheses, the name of the original
author or publisher of that section if known, or else a unique number.
Make the same adjustment to the section titles in the list of
Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History"

in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright (c) YEAR YOUR NAME.
> Permission is granted to copy, distribute and/or modify this document
> under the terms of the GNU Free Documentation License, Version 1.3
> or any later version published by the Free Software Foundation;
> with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
> A copy of the license is included in the section entitled "GNU
> Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

> with the Invariant Sections being LIST THEIR TITLES, with the
> Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.