Term49 Shell Documentation

This file gives an overview of the Term49 shell, and is broken down into topics:

==========

1. QUICK START

You can access the various special keys (Ctrl, Alt, arrows, etc.) by touching the top left corner of the screen, and then pressing special keys on the keyboard. When you touch the top left corner of the screen, you will enter 'metamode', which remaps the keyboard keys to other functions. You can tell when you are in metamode by the green 'M' in the top right corner of the screen. Touching the top left corner of the screen will toggle metamode on and off.

The default metamode keys are:

h: left arrow
j: down arrow
k: up arrow
l: right arrow
t: tab
e: escape
c: control
a: alt
v: paste

After pressing a key in metamode you will be returned to normal input, excepting the arrow keys, which leave you in metamode so you can move the cursor around easily.

For example, if you wanted to type Ctrl-C (^C), then you would touch the top left corner of the screen to enter metamode, press 'c' to hold down control, and then 'c' again to send ^C. If you wanted to move the cursor left, you would touch the top left corner of the screen to enter metamode, then 'h' to move the cursor left (perhaps several times). Once you have the cursor where you want it, you can start typing again by touching the top left corner of the screen (exiting metamode). Typing any key that is not bound to a metamode function will simply exit metamode.

If you have a physical keyboard on your device, you can hold the space key to toggle metamode on, and also hold down ordinary keys to get uppercase letters.

If your device has a physical shift key, or you are using a bluetooth keyboard, you can also doubletap the right shift key to toggle metamode on and off. Note that some keyboards may not send right shift when you press the right shift key (it may send left shift), but this can be addressed in preferences.

The metamode keys are all configurable, as described below.

==========

2. PREFERENCES

Preferences are read from the .term49rc file in $HOME. If this file does not exist at startup it will be created at launch time with the default settings (along with this README). Term49 uses libconfig for preferences, so the syntax of the .term49rc file is as per libconfig. If there is an error reading the preference file it will be ignored and Term49 will launch with the defaults.

Each of the preference options and their default is shown below:

```
font_path = "/usr/fonts/font_repository/monotype/andalemo.ttf";
/* The path to the Unicode TTF font file to use in the
 * shell. This can be an absolute or relative path. */

font_size = 16;
/* The font size for the shell. Term49 will try to guess a
 * reasonable size for your screen, but otherwise
 * defaults to this value. */

text_color = [255, 255, 255];
/* The color for text, in RGB format. If you would like
 * classic green text on black background, you can change
 * this to [0, 255, 0]. */

background_color = [0, 0, 0];
/* The background color, in RGB format. */

screen_idle_awake = false;
/* If set to true, Term49 will force the screen to stay
 * awake when idle instead of dimming and eventually
 * turning off as usual. */

auto_show_vkb = true;
/* When set to true, will automatically display the virtual
 * keyboard at launch time unless you have a Bluetooth
 * keyboard. Set to false to disable. On a Passport, this
 * is always true so the symbol keys can be accessed. */

metamode_doubletap_key = 61666;
/* Which key to double tap in order to toggle metamode on
 * and off. This is an integer corresponding to the key
 * symbol and defaults to KEYCODE_RIGHT_SHIFT.  */

metamode_doubletap_delay = 500000000;
/* When double tapping to invoke metamode, the time in
 * which to double tap, in nanoseconds. Defaults to 0.5
 * seconds. */

keyhold_actions = true;
/* Enables special actions when holding down a key. By
 * default, Term49 will upcase letters if you hold them, and
 * will toggle metamode on if you hold space. */

keyhold_actions_exempt = [KEYCODE_BACKSPACE,KEYCODE_RETURN];
/* These keys do not trigger special handling when held
 * down, so they will repeat as usual. */

metamode_hold_key = 32;
/* If keyhold_actions is true, then holding this key will
 * toggle metamode on. Defaults to KEYCODE_SPACE. */

tty_encoding = "UTF-8";
/* Term49 uses 16 bit Unicode internally, but when doing IO
 * with the tty / shell, will convert between Unicode and
 * the tty_encoding. Defaults to UTF-8, but can be any
 * encoding that the International Components for Unicode
 * (ICU) understands. */

metamode_hitbox : { x = 0; y = 0; w = 100; h = 100; };
/* This is the area of the screen that you touch to toggle
 * metamode on and off, specified from the top left corner
 * of the screen. Defaults to a 100x100 pixel box in the
 * top left corner, but you can move it somewhere else if
 * you like by specifying the top left corner of the box (x
 * from the left, y from the top) and the width (w) and
 * height (h) in pixels. */
```

```
metamode_keys : (
    ("e", "\x1B"),
    ("t", "\t")
  );
/* When in metamode, pressing the given key will send the
 * specified string into the shell and then turn off
 * metamode. Strings are read according to the rules for
 * string parsing in libconfig. You can specify hex
 * characters using \x and exactly two characters. Term49
 * uses this mechanism to send control sequences into the
 * shell, but you could specify any character string you
 * want, including UTF-8 sequences.
 *
 * Term49 supports terminfo aliases for some common escape
 * sequences. Specifically, if a metamode key is set to
 * one of kcuu1, kcud1, kcuf1, kcub1, khome, kend, or
 * kf1 through kf12, then the correct escape sequence will
 * be sent for cursor up, down, right, left, home, end or
 * the corresponding function key. */

metamode_sticky_keys : (
    ("k", "kcuu1"),
    ("j", "kcud1"),
    ("l", "kcuf1"),
    ("h", "kcub1")
  );
/* Like metamode_keys above, but leaves metamode on
 * afterwards, so you can press another metamode key
 * immediately. */

metamode_func_keys : (
    ("a", "alt_down"),
    ("c", "ctrl_down"),
    ("s", "rescreen"),
    ("v", "paste_clipboard")
  );
/* These keys map to special functions in Term49:

  alt_down: Will 'hold down' the alt key and send it as a
  modifier with the next keypress.

  ctrl_down: Will 'hold down' the ctrl key.

  rescreen: Will reinitialize the screen surface and fonts,
  including font size. You may want this is you have
  allow_resize_columns = true.

  paste_clipboard: Will paste the contents of the system
  keyboard. Handy for pasting URLs into the shell. Note
  that no character encoding or conversion will be done.
*/

rescreen_on_symmenu = true;
/* If rescreen_on_symmenu is true, then opening a symbol
  menu will cause the screen to readjust to new dimensions
*/

main_symmenu : ( (
    ("a", "="),
    ("s", "-"),
    ("d", "*"),
    ("f", "/"),
    ("g", "\\"),
    ("h", "|"),
    ("j", "&"),
    ("k", "'"),
    ("l", "\"")
  ),
  (
    ("q", "~"),
    ("w", "`"),
    ("e", "{"),
    ("r", "}"),
    ("t", "["),
    ("y", "]"),
    ("u", "<"),
    ("i", ">"),
    ("o", "^"),
```

```
    ("p", "%")
  ) );
/* For devices with a Sym key, this preference controls the
 * contents of the symbol menu. */

sticky_sym_key = false;
/* For devices with a Sym key, controls whether the Sym
 * menu stays open after hitting a key. */

sticky_shift_key = false;
/* For devices with physical shift keys, this setting
 * controls whether the shift key 'sticks' after being
 * pressed, and applies to the next keypress. */

allow_resize_columns = false;
/* This setting controls whether a program can change the
 * number of columns on the screen. Defaults to false (no
 * resizing). When set to true, programs can set the number
 * of columns (80 or 132), but this may make the text
 * really small. Use the metamode 'rescreen' function to
 * reset the font size to your preference. */

prefs_version =
/* This is the current version of the preferences file,
 * according to Term49. If it is different than the app
 * version, then Term49 will update the preferences file to
 * the new version. You don't need to worry about this. */
```

==========

3. WHAT'S INSTALLED

The default userland that comes with your device is pretty sparse, and some of the versions are somewhat old. You have the basic file utilities (ls, less, mv, cp, etc.), but no compiler and no man pages. Some commands have 'use' information available (eg. 'use /bin/ls') which can help out with the flags for the QNX userland tools. You do have python so you can get some work done with that. You can edit files with elvis, which is like vi.

The default version of ksh is quite old, so Term49 bundles mksh, which is better in general and much better about unicode in particular. Term49 also bundles ssh, scp, sftp and ssh-keygen, due to popular demand.

It is possible to install other programs and round out the userland, as described below in FOR DEVELOPERS.

==========

4. FOR DEVELOPERS

Term49 is Open Source

Term49 is a 'from scratch' terminal emulator in that it is not a port of another program. It uses SDL to draw to the screen, freetype to render fonts, libconfig to manage preferences, ICU to convert between character encodings, and implements (parts of) the xterm specification. It is not by any means perfect, and will certainly have bugs or missing features. Because the author enjoyed access to so many high quality libraries that made the development of Term49 significantly easier, Term49 is open source:

https://github.com/mordak/Term49

If you find a bug, missing feature, or something else, feel free to bring it up on github. Pull requests are welcome.

Adding Programs to the Userland

Because you have python and sh, and BlackBerry provides a cross compiler with their dev tools, it is possible to cross compile programs and install them into your userland. Because the dev tools use gcc, it is possible to cross compile gcc and install that into your userland, and then you can compile other programs locally. This is not trivial, but is also not too hard. The author has a very rudimentary bootstrap system that will cross compile gcc and some other useful tools and then install them over the air onto your device - no root required. You can give it a try if you like:

https://github.com/mordak/playbook-dev-tools

This has been tested with a Linux (ubuntu) build machine and a Passport target. Again, feel free to use github to contribute or raise issues. In the long term the goal is to port some kind of existing package management system and bundle it with Term49, but we're not there yet.

Happy hacking!

==========

5. ALTERNATIVE .term49rc CONFIGURATIONS

Below are several alternative configuration files optimized for the BlackBerry Passport (1440x1440 display) with different themes and the Source Code Pro font.

5.1 Solarized Dark Theme (Recommended)

```
// Solarized Dark .term49rc for BlackBerry Passport
// Optimized for 1440x1440 display with Source Code Pro font
// Save this as .term49rc in your $HOME directory

font_path = "/usr/fonts/truetype/sourcecodepro/SourceCodePro-Regular.ttf";
/* Source Code Pro font - excellent for coding and terminal use
 * Alternative paths if not found:
 * "/system/fonts/SourceCodePro-Regular.ttf"
 * "/usr/share/fonts/truetype/sourcecodepro/SourceCodePro-Regular.ttf"
 */

font_size = 18;
/* Optimized for Passport's 1440x1440 display - about 40 chars wide */

text_color = [131, 148, 150];
/* Solarized base0 - main text color */

background_color = [0, 43, 54];
/* Solarized base03 - dark background */

screen_idle_awake = true;
/* Keep screen awake for coding sessions */

auto_show_vkb = true;
/* Always true on Passport for symbol access */

metamode_doubletap_key = 61666;
/* Right shift double-tap */

metamode_doubletap_delay = 500000000;
/* 0.5 second delay */

keyhold_actions = true;
/* Enable key hold actions */

keyhold_actions_exempt = [KEYCODE_BACKSPACE,KEYCODE_RETURN];
/* Standard exempt keys */

metamode_hold_key = 32;
/* Space for metamode toggle */

tty_encoding = "UTF-8";
/* UTF-8 for full Unicode support */

metamode_hitbox : { x = 0; y = 0; w = 120; h = 120; };
/* Sized for Passport touch precision */

metamode_keys : (
    ("e", "\x1B"),
    ("t", "\t")
  );

metamode_sticky_keys : (
    ("k", "kcuu1"),
    ("j", "kcud1"),
    ("l", "kcuf1"),
    ("h", "kcub1")
  );

metamode_func_keys : (
    ("a", "alt_down"),
    ("c", "ctrl_down"),
    ("s", "rescreen"),
    ("v", "paste_clipboard")
  );
```

```
rescreen_on_symmenu = true;

main_symmenu : ( (
    ("a", "="),
    ("s", "-"),
    ("d", "*"),
    ("f", "/"),
    ("g", "\\"),
    ("h", "|"),
    ("j", "&"),
    ("k", "'"),
    ("l", "\"")
  ),
  (
    ("q", "~"),
    ("w", "`"),
    ("e", "{"),
    ("r", "}"),
    ("t", "["),
    ("y", "]"),
    ("u", "<"),
    ("i", ">"),
    ("o", "^"),
    ("p", "%")
  ) );

sticky_sym_key = false;
sticky_shift_key = false;
allow_resize_columns = false;
prefs_version = 1;
```

5.2 Matrix Theme - Classic (Green on Black)

```
// Matrix Theme .term49rc for BlackBerry Passport
// Classic green-on-black hacker aesthetic
// Save this as .term49rc in your $HOME directory

font_path = "/usr/fonts/truetype/sourcecodepro/SourceCodePro-Regular.ttf";
/* Source Code Pro for that authentic code feel */

font_size = 18;
/* Optimized for Passport display */

text_color = [0, 255, 65];
/* Bright Matrix green - #00FF41 */

background_color = [0, 0, 0];
/* Pure black background */

screen_idle_awake = true;
/* Stay awake for extended hacking sessions */

auto_show_vkb = true;

metamode_doubletap_key = 61666;
metamode_doubletap_delay = 500000000;
keyhold_actions = true;
keyhold_actions_exempt = [KEYCODE_BACKSPACE,KEYCODE_RETURN];
metamode_hold_key = 32;
tty_encoding = "UTF-8";

metamode_hitbox : { x = 0; y = 0; w = 120; h = 120; };

metamode_keys : (
    ("e", "\x1B"),
    ("t", "\t")
  );

metamode_sticky_keys : (
    ("k", "kcuu1"),
    ("j", "kcud1"),
    ("l", "kcuf1"),
    ("h", "kcub1")
  );

metamode_func_keys : (
```

```
    ("a", "alt_down"),
    ("c", "ctrl_down"),
    ("s", "rescreen"),
    ("v", "paste_clipboard")
  );

rescreen_on_symmenu = true;

main_symmenu : ( (
    ("a", "="),
    ("s", "-"),
    ("d", "*"),
    ("f", "/"),
    ("g", "\\"),
    ("h", "|"),
    ("j", "&"),
    ("k", "'"),
    ("l", "\"")
  ),
  (
    ("q", "~"),
    ("w", "`"),
    ("e", "{"),
    ("r", "}"),
    ("t", "["),
    ("y", "]"),
    ("u", "<"),
    ("i", ">"),
    ("o", "^"),
    ("p", "%")
  ) );

sticky_sym_key = false;
sticky_shift_key = false;
allow_resize_columns = false;
prefs_version = 1;
```

5.3 Matrix Theme - Reverse (Black on Green)

```
// Matrix Reverse Theme .term49rc for BlackBerry Passport
// Black text on green background - unique reverse Matrix look
// Save this as .term49rc in your $HOME directory

font_path = "/usr/fonts/truetype/sourcecodepro/SourceCodePro-Bold.ttf";
/* Bold Source Code Pro for better contrast on green background */

font_size = 18;
/* Passport-optimized size */

text_color = [0, 0, 0];
/* Black text */

background_color = [0, 255, 65];
/* Matrix green background - #00FF41 */

screen_idle_awake = true;
/* Keep screen active */

auto_show_vkb = true;

metamode_doubletap_key = 61666;
metamode_doubletap_delay = 500000000;
keyhold_actions = true;
keyhold_actions_exempt = [KEYCODE_BACKSPACE,KEYCODE_RETURN];
metamode_hold_key = 32;
tty_encoding = "UTF-8";

metamode_hitbox : { x = 0; y = 0; w = 120; h = 120; };

metamode_keys : (
    ("e", "\x1B"),
    ("t", "\t")
  );

metamode_sticky_keys : (
    ("k", "kcuu1"),
```

```
    ("j", "kcud1"),
    ("l", "kcuf1"),
    ("h", "kcub1")
  );

metamode_func_keys : (
    ("a", "alt_down"),
    ("c", "ctrl_down"),
    ("s", "rescreen"),
    ("v", "paste_clipboard")
  );

rescreen_on_symmenu = true;

main_symmenu : ( (
    ("a", "="),
    ("s", "-"),
    ("d", "*"),
    ("f", "/"),
    ("g", "\\"),
    ("h", "|"),
    ("j", "&"),
    ("k", "'"),
    ("l", "\"")
  ),
  (
    ("q", "~"),
    ("w", "`"),
    ("e", "{"),
    ("r", "}"),
    ("t", "["),
    ("y", "]"),
    ("u", "<"),
    ("i", ">"),
    ("o", "^"),
    ("p", "%")
  ) );

sticky_sym_key = false;
sticky_shift_key = false;
allow_resize_columns = false;
prefs_version = 1;
```

Installation and Font Setup

Font Installation:

- Download Source Code Pro from Adobe Fonts or Google Fonts
- Copy font files to `/usr/fonts/truetype/sourcecodepro/` or `/system/fonts/`
- If Source Code Pro is not available, the system will fall back to the default monospace font

Configuration Installation:

- Choose one of the three configurations above
- Copy the entire configuration to a new file named `.term49rc`
- Place this file in your home directory (`$HOME`)
- Restart Term49 to apply the new configuration
- Use metamode 'rescreen' function if display adjustment is needed

BlackBerry Passport Optimizations:

- **Font Size 18:** Perfect balance for the 1440x1440 display - approximately 40 characters wide
- **120x120 hitbox:** Sized for precise touch interaction on Passport
- **Column resizing disabled:** Prevents text from becoming too small
- **Screen always awake:** Ideal for development and long terminal sessions

Theme Descriptions:

- **Solarized Dark:** Professional, eye-friendly theme perfect for extended coding sessions
- **Matrix Classic:** Iconic green-on-black for that authentic hacker aesthetic

- **Matrix Reverse:** Unique black-on-green theme that stands out and reduces eye strain in bright environments