```
12/13/20 06:46:06 Z:\_shares1\psystems\PBM4Y-RASPB\BM4Y-DS1-DigiSign.yaml
```

                            BM4Y-DS1
               OpiBerry Embedded ARM Cortex-A72/A53
               Software Guide / OpiDigi Signage Player


BM4Y-DS1 OpiDigi Signage Player
   Linux based built-in Chromium Brower
   Play resources from Internet, local network or local folders
   Display more attractive resources, easily
[BM3BP-hwfrontb1.jpg ; BM3BP-hwfrontw1.jpg ; BM4Y-DS1-DS1.jpg]


Copyright © Opiware Solutions. All Rights Reserved. | Version: 0.1  2020 Nov.


                                          BM4Y-DS1 Software Guide
-------------------------------------------------------------------------

Trademarks
The Opiware logo is a registered trademark of Opiware Solutions. All other
trademarks or registered marks in this manual belong to their respective
manufacturers.

Disclaimer
Information in this document is subject to change without notice and does not
represent a commitment on the part of Opiware.

Opiware provides this document as is, without warranty of any kind, either
expressed or implied, including, but not limited to, its particular purpose.
Opiware reserves the right to make improvements and/or changes to this manual, or
to the products and/or the programs described in this manual, at any time.

Information provided in this manual is intended to be accurate and reliable.
However, Opiware assumes no responsibility for its use, or for any infringements on
the rights of third parties that may result from its use.

This product might include unintentional technical or typographical errors. Changes
are periodically made to the information herein to correct such errors, and these
changes are incorporated into new editions of the publication


                                          BM4Y-DS1 Software Guide
-------------------------------------------------------------------------


                         Document Amendment History
   Revision     Date         Remark
   V 0.1        2020 Nov.    Initial Doc

BM4Y-DS1 Software Guide
---------------------------------------------------------------------------


Table of Contents

BM4Y-DS1 Software Guide
---------------------------------------------------------------------------


1. Overview
OpiWebUI based Digital Signage is an operating system built-in Chromium Brower GUI that is designed to always run fullscreen restricted to a specified Single App Kiosk Mode do not allow the user to exit the app. They're great for a purpose-built device, such as this OpiDigi Signage Player.

It shows web resources from Internet, local network or local folders (in RPi itself as the source webserver. It's up to user to configure the internal web page). Application usages are, use it display TV advertising screen, electronic signs, booking site, queue or timetable management web app or create superb web presentations with Youtube, Google Slides (Powerpoint compatible) or similar software.

OpiWebUI comes with the latest Chromium builds (featuring HTML5/CSS3/JavaScripy capabilities), so you can display more attractive resources, more easily.

2. Connect to the Network

2.1 The Linux OpiWebUI OS is pre-installed and shippped with the RPi hardware assembled. When you first plug in power to the RPi, it will show a black screen and then some text will appear as it boots up. If the RPi connects to your Internet router using the Ethernet wired cable, it will show some text of network diagnostics info, take note of this page as it will display your IP address, which you will need later.


-----------------------------------------
[RDS-netinfo.png ;Network Info screen]
eth0 ; lo ; wlan0
-----------------------------------------


2.2 Network admin interface: If no wired network is available, it boots up showing the Network Settings page, login using the default login [user:admin/pswd:admin123], where you can setup your network connection wired or

wireless to access to the RPi. Once you select a connection method, perform reboot to take effect.

```
-----------------------------------------
[RDS-netset.png ;Network Settings screen]
Wired network (Ethernet)
Wireless networks (802.11) near you
-----------------------------------------
```

Wireless networks: It should list out the available WIFI networks. Select your network, then enter the password and click connect. Note: You might have to click connect and let it run twice.

2.3 Admin interface: If wired or wireless network is available, it boots up in full-screen kiosk mode displaying the configured web resource. System's IP address is shown on screen upon boot. To configure the system, browse from your PC to <http://<IP_ADDRESS>/settings>. Once network is set, administration can/must be performed remotely via browser, without physical access to the RPi. You can always modify RPi's network connection by browsing from your PC to <http://<IP_ADDRESS>/networks>.

```
-----------------------------------------
[RDS-sigset.png ;Signage settings screen]

Chromium settings
Make all Chromium settings <persistent>: all modifies you apply to Chromium will be
saved (browser configuration requires physical/VNC access to the Pi). Toggle to
<default settings> again.

Admin interface password
Change the <password> for this admin interface. This will also change the password
of the login for the rds-user (SSH, too) and the VNC password.

Kiosk settings
* Web resource's URL: https://www.opiware.com/
* Reset browser after user inactivity of [minutes]:
* Force reloading of web page every [seconds]     :
* Set HTTP proxy [example 10.11.12.13:80]         :
* Halt system at [UTC 24h, hh:mm]                 :
<SAVE AND START KIOSK>
-----------------------------------------
```

Chromium settings: While you can configure the system via the aforementioned settings web page quite completely, when you need to modify browser's settings and extensions a virtual access to the whole RPi's screen is required: just browse to the settings page via browser and then access the RPi by using VNC default [password:admin123]. This way you can customize the browser as you wish.
* In order for your settings to be saved, make all Chromium settings persistent. Why this?
Every time system reloads or is rebooted, the browser is kept back to default settings or last-saved settings, which are "clean", for security reasons.

Admin interface password: This function changes the admin interface password, as

well as the rds-user password for SSH and the VNC one all together. Remember to change the default password at first setup.

Kiosk settings: Here you can configure the URL of the web resource displayed by the full-screen browser.
* Type in the URL for the RPi to load when it boots up. As far as the kiosk type, choose the "Full screen view."
* A token (the machine's MAC address without ":" characters) is transparently added at the very end of the web resource URL, allowing multiple Pis pointing just one server location, for example <http://yourserver.com?rdstoken=080027fe959b>. It's then up to your server logic sending the appropriate resource to the client, if you need different content for each client to be displayed.
* A browser full reset after a specified user inactivity feature is available, which will – nomen omen – kill and completely reload the browser, also cleaning up all older content and replacing it with the last-saved.
* A reloading of web page content after a specified time is possible as well. Please note that this is a hack (normally it's not possible to interfere with web pages' behaviour from the outside of them) but quite all sites are however compatible.

Some default Chromium extensions are installed:
* Scrollbar Customizer <https://Chromium.google.com/webstore/detail/scrollbar-customizer/flffekjijpabhjgpoapooggncnmcjopa> (enabled): with this extension it's possible to modify the scrollbars' sizes, thick to hidden. Very thin scrollbars are set as default.
* Virtual keyboard <https://Chromium.google.com/webstore/detail/virtual-keyboard/pflmllfnnabikmfkkaddkoolinlfninn>, on-screen virtual keyboard (enabled).
* tabtiles <https://Chromium.google.com/webstore/detail/tabtiles/aaeapgfkbbbdpbfjmpcblemfajmkiddh> (disabled). Enable this if you need to display a useful pseudo-navigation menu (if the displayed resource opens window popups).
* Url slideshow <https://Chromium.google.com/webstore/detail/url-slideshow/pdblffiahfjjldpkngdpaegghhamefam> (enabled). Configure it to turn the signage in a slideshow of web sites (check the extension's options Start on browser start and Fullscreen).

-----------------------------------------
[RDS-sysset.png ;System settings screen]

Locale
Set browser locale and keymap: [English (US) (default)]

Video Resolution
View screen resolutions: show available (not all resolutions are usable by the Pi).
Set resolution: [1824 x 984] | <save and use.>

Zoom
Set zoom (for HiDPI screens): [Select zoom level...]

Video Rotation
Rotation: (*)normal ()left ()right ()reverse | <apply> (system will be rebooted).
-----------------------------------------

Kiosk mode: OpiWebUI boots in full-screen (kiosk) mode displaying the saved web resource with all the saved settings applied; mouse pointer auto-hides in some seconds of inactivity. For security reasons, some keyboard keys are disabled.

You cannot exit to the console in any way except stopping the CDS service via SSH. Please do not hard reboot your RPi while in production.

Kiosk mode with visible address bar: OpiWebUI is designed for digital signage installations – it's a full-screen browser-face system. The <Web Kiosk plugin> adds an address bar and tunes the system in a way it can be used for "web workstations" (often found in cafès, offices, schools, hotels, hospitals, libraries), where people can freely surf the web; a stylish "address bar" becomes available, even if the browser is in full-screen mode.

3. Configuration

Using SSH: You can SSH into the RPi in order to set up the underlying Raspbian operating system (usage is similar to standard Debian OS) and configure OpiWebUI without a GUI as well.

3.1 SSH: Perform an SSH login with the following clients:
*nix users will make use of the native ssh client – open the terminal emulator and type:
~$ ssh rds-user@IP_ADDRESS

3.2 Windows users will use Putty, MobaXTerm, or any similar program. Use the following default credentials: [user:rds-user/password:admin123] Once in, type:
~$ sudo -i  ;for administrative rights (root)

When admin screen password is modified, SSH password will be changed accordingly. So, if you set a password for the admin interface, the new SSH password will be identical. This way you can protect both system settings' modification via browser (admin interface) and via SSH with one step. Please change the default SSH password for your security.

3.3 Configure the system via SSH
Modify the config file of your interest in the /etc/rds/ folder and then restart the OpiWebUI service: ~# systemctl restart rds

* For the network, standard /etc/network/interfaces.d/* and /etc/resolv.conf files are used.

* For the locale and keyboard, the standard system files /etc/default/locale and /etc/default/keyboard are used.

* For the rotation, the /boot/config.txt file is used by the OpiWebUI stack.

4. Plugins: See the plugins page in following.

4.1 Web Kiosk plugin: This plugin adds an address bar to OpiWebUI and tunes it in a system suitable for usage in web kiosks and multi-user web workstations environments (cafès, offices, schools, hotels, hospitals, libraries), where people

can freely surf the web. OpiWebUI with the Web Kiosk plugin replaces the older RPi
WebKiosk.

Example images (click to view) [RDS-wk-webkiosk1.png ; RDS-wk-webkiosk2.png]

Requirements: This plugin requires OpiWebUI >= v.13.1

Install: From a SSH terminal of OpiWebUI, as root:
~# cd /root/libtools/
~# dpkg -i rds-plugin-webkiosk*.deb

Uninstall:
~# apt --purge remove -y rds-plugin-webkiosk*

4.2 WordPress plugin: OpiWebUI admin interface allows you to type in the URL of the
web resource to be displayed; it can be an Internet one (https://www.opiware.com),
a LAN URL (http://192.168.1.28/booking; http://booking.lan), or even a resource
located internally, inside the Raspberry Pi's filesystem.

To simplify the management of the internal site setup, the wordpress plugin adds a
WordPress installation to the system (/rds/var/www/wordpress).

[RDS-wp-hello.png]

Requirements: This plugin requires OpiWebUI >= v.13.1

Install: From a SSH terminal of OpiWebUI, as root:
~# apt update
~# cd /root/libtools/
~# dpkg -i rds-plugin-wordpress*.deb ; ~# apt install -f -y

Depending on the packages installed, a different php-mysql package could be
required:
~# apt install -y php7.1-mysql
~# systemctl restart apache2
~# systemctl restart rds

Prerequisites: In order to view (and administer) the RPi's WordPress site from your
PC, just add the following line to your PC hosts file:
RaspberryPi_LAN_IP  wordpress
For example: 192.168.0.28  wordpress

This way, your PC will identify the URL <http://wordpress> as coming from the Pi.
Google for what "hosts" file means if unsure.

Usage: Set up your site on the internal WordPress installation: as a normal
WordPress site, by browsing from your PC to <http://wordpress>

WordPress admin login is: [admin/password]
Remember to change it. Opiware won't give basis/informations on how to use
WordPress.

Make OpiWebUI display it: from the admin interface, set <http://wordpress> as the

kiosk URL.

Disable network check: Upon boot, the system checks for the availability of the network, in order to display the proper admin interface if no network is available. However, with an internal WordPress installation, the network check could be unwanted: in case, modify /rds/home/rds-user/.xinitrc as:

```
connectionCheck()
{
    return 0
}
```

Browsing whitelist: In order to add an internal filtering HTTP proxy with a whitelist behaviour (users allowed to browse only a specific set of sites), use follow the steps.

4.2.1 Install tinyproxy via apt and change its config file to include:
FilterURLs On
FilterDefaultDeny Yes
Filter "/etc/tinyproxy/whitelist"

4.2.2 Create a whitelist with only the domains that you want the user to browse (when displaying your websites, make sure no resource loading is blocked by the whitelist; use the Chromium's developer console for the purpose).

4.2.3 Set localhost:8888 as the proxy server in the admin interface.

Uninstall
~# apt --purge remove rds-plugin-wordpress*
~# rm -R /rds/var/www/wordpress/wp-content

5. FAQ: All Frequently asked question.