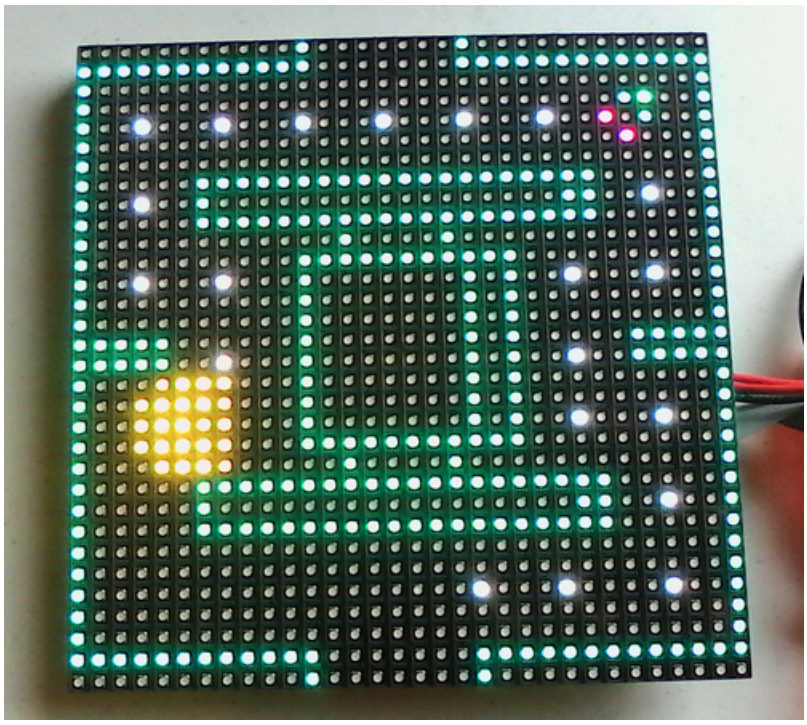
adafruit learning system

Web Enabled PIXEL on Raspberry Pi

Created by Roberto Marquez



Last updated on 2018-08-22 03:45:27 PM UTC

Guide Contents

Overview

The PIXEL kit from LED:ART consists of an RGB LED matrix, a IOIO Mint Bluetooth microcontroller to drive the matix, and some software to give instructions to the IOIO for what to display. (https://adafru.it/ejZ)

Originally, only an Android and PC based software was available. Now, web based controls are available, via a Java application. Web based controls means that platforms with a modern browser are supported; iStuff, Android, PC, Tablet, and others. The Web applicaion is started from the command line.

Being able to run from the command line allows for use as a background service/daemon. This approch is also ideal for setting up headless envronments (no monitor attached to the mother board), where the hardware profile needs to be as slim as possible.

PIXEL comes pre-bundled with some still and animated images. On first run, the images are extracted to pixel/ under the user's home directory.

Once the hardware is assembled, and the software running, the PIXEL Web controls are availble to any computer on the same network as the Raspberry Pi.

Here is a video showing some of the available animations:

Hardware

Required Hardware

- Raspberry Pi (http://adafru.it/1914)
- USB A-Male to A-Male cable
- Pixel Guts (http://adafru.it/1357)

Hardware for Networking (one or the other)

- WiFi Module (http://adafru.it/814)
- Ethernet Cable (http://adafru.it/730)

Optional Hardware for Wearables

To go wearable/portable, power the Raspberry Pi and Pixel board with two lipo batteries and two Adafruit PowerBoots and corresponding USB adaptors:

- PowerBoost (http://adafru.it/2030)
- LiPo Batteries (http://adafru.it/328)
- USB to USB Micro B cable (http://adafru.it/1513) (for powering the Raspberry Pi)
- USB to Barrel Jack Adapter (for powering the LED matrix)

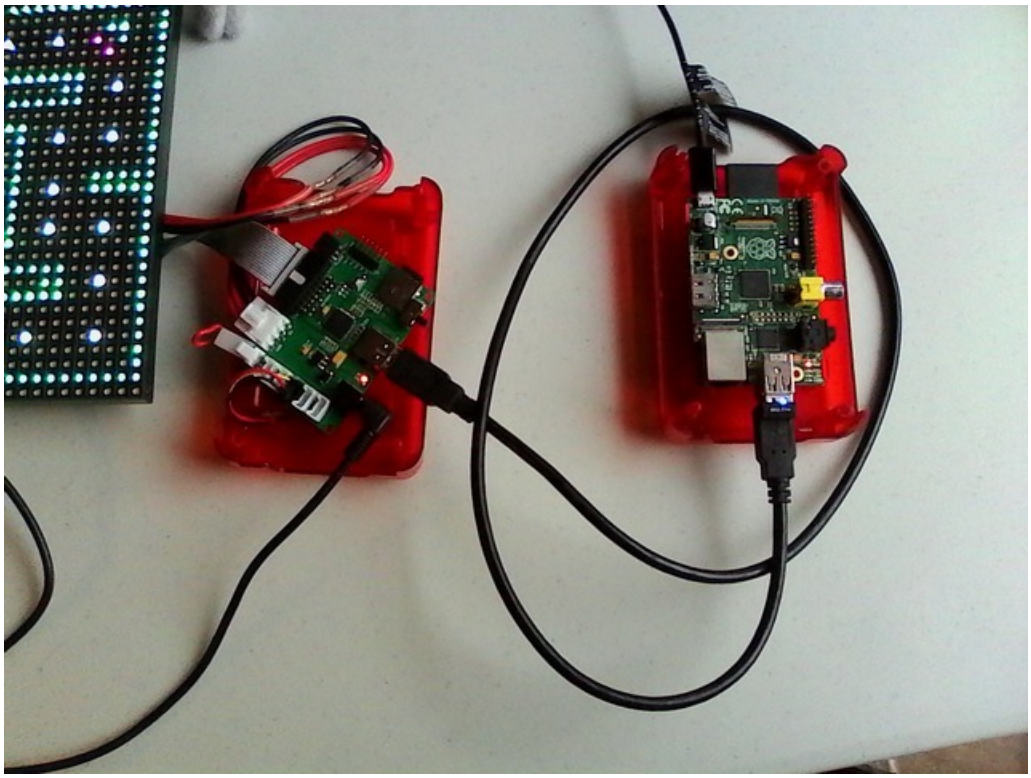
Note: The Pixel started off as a Kickstarter which is version 1 of the product. Some version 1 Pixels were even in the Adafruit store. The Pixel software used in this guide is compatible with both the version 1 and version 2 models.

Assembly

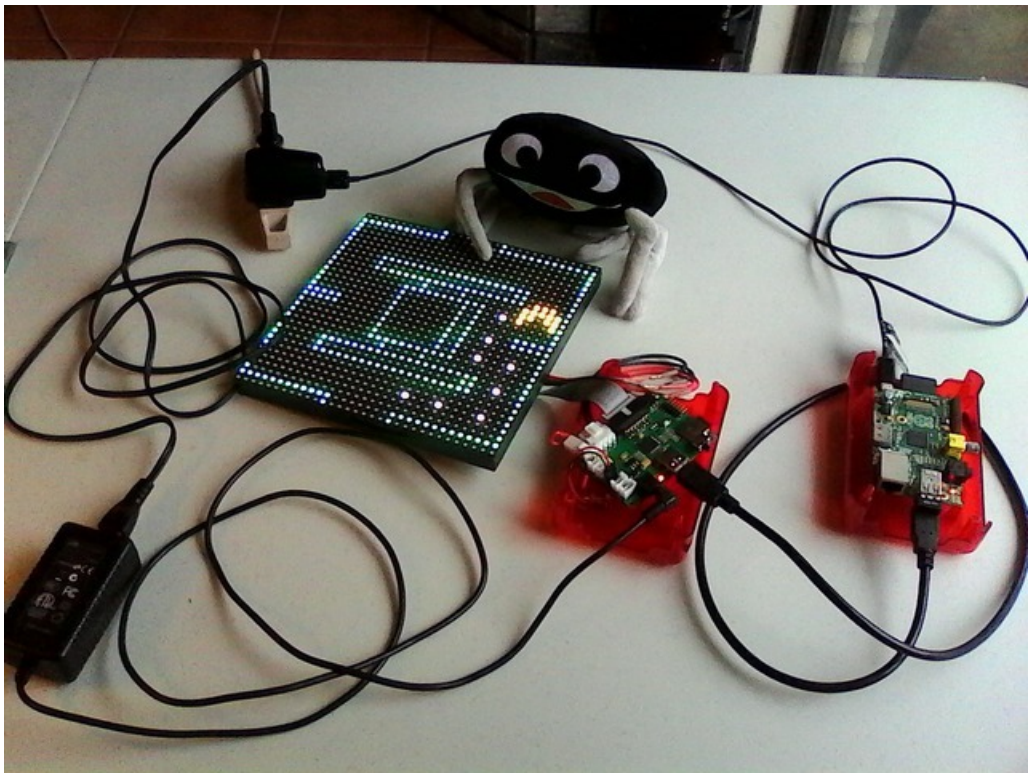
See the following links to get your Raspberry Pi and Pixel setup and connected to power, but don't give them power just yet.

- Lean about using your Raspberry Pi (https://adafru.it/dpe)
- Read the PIXEL Guts (https://adafru.it/emB) Quickstart guide

The last assembly step is to connect the Raspberry Pi to the Pixel, via a male-A to male-A USB cable.



Now power up the Raspberry Pi and Pixel.



Software

The target platform for this project is Debian Linux; Raspbian/Ubuntu.

The application runs on Raspberry Pi and PC.

The Pixel software used in this guide is distributed as a executable JAR file (Java Archive). This means a recent version of the JVM is needed.

If you are using the lastest version of Raspbian, then a suitable JVM is already installed.

If you don't have at least Java 8 on the Raspberry Pi, then update and install it.

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install oracle-java8-jdk
```

Next, download the executable JAR file.

https://adafru.it/fw8

https://adafru.it/fw8

Now send the JAR to the Raspberry Pi, using your favorite file transer method.

The **scp** command is an option for file transfer:

```
scp pixel.jar pi@raspberrypi.local:
```

or you can try from the Pi itself

wget https://learn.adafruit.com/system/assets/assets/000/026/192/original/pixel-web-enabled-0.0.1-SNAPSHOT-jar-with-dependencies.jar?1435773492

Usage

Starting the App

Run the Web application with the following command.

```
sudo java -jar pixel.jar
```

In the above command, 'pixel.jar' should be replaced with the actual name of the JAR file copied to the Raspberry Pi.

There are some command line switches to configure the application. See below for details.

Once the Java application is running from the command line, a bunch of lines scroll on the terminal showing connection messages. If it is the first run, then image resource extraction messages are also shown.

Next, visit the following URL.

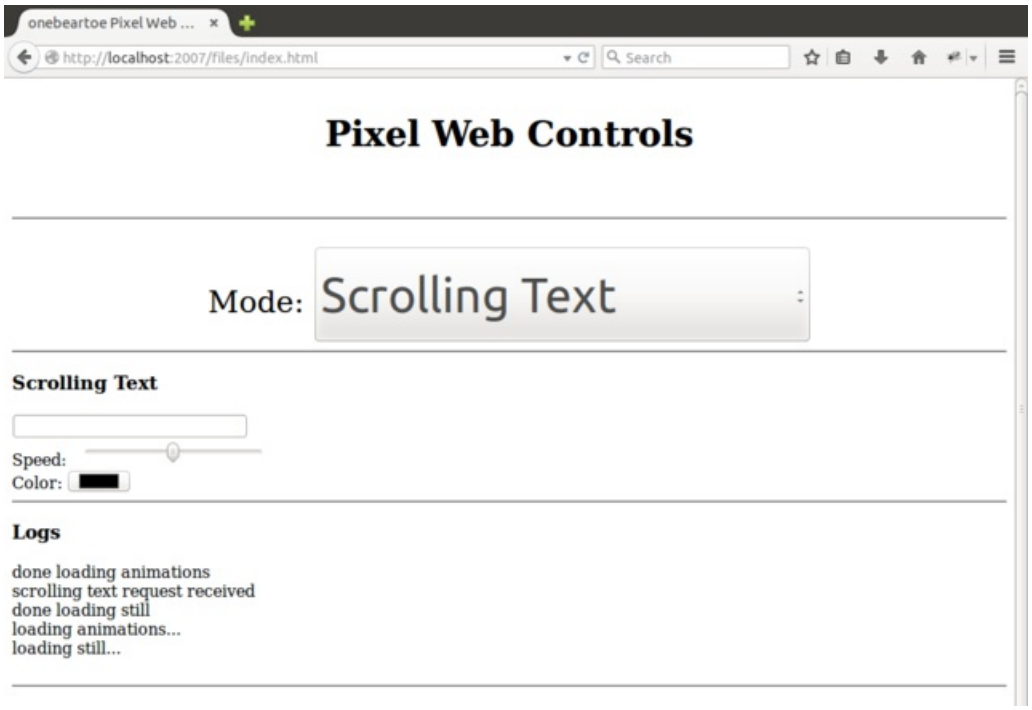
http://raspberrypi:8080/files/index.html (https://adafru.it/fvF)

The 'raspberrypi' in the above URL should be replaced with the host name (or IP address) of computer running the Pixel Web applicaton.

At the above URL, a user interface is presented with a drop down menu to change between Pixel modes.

Scrolling Text

The scrolling text mode lets you enter some text, change scroll speed, and specify text color.

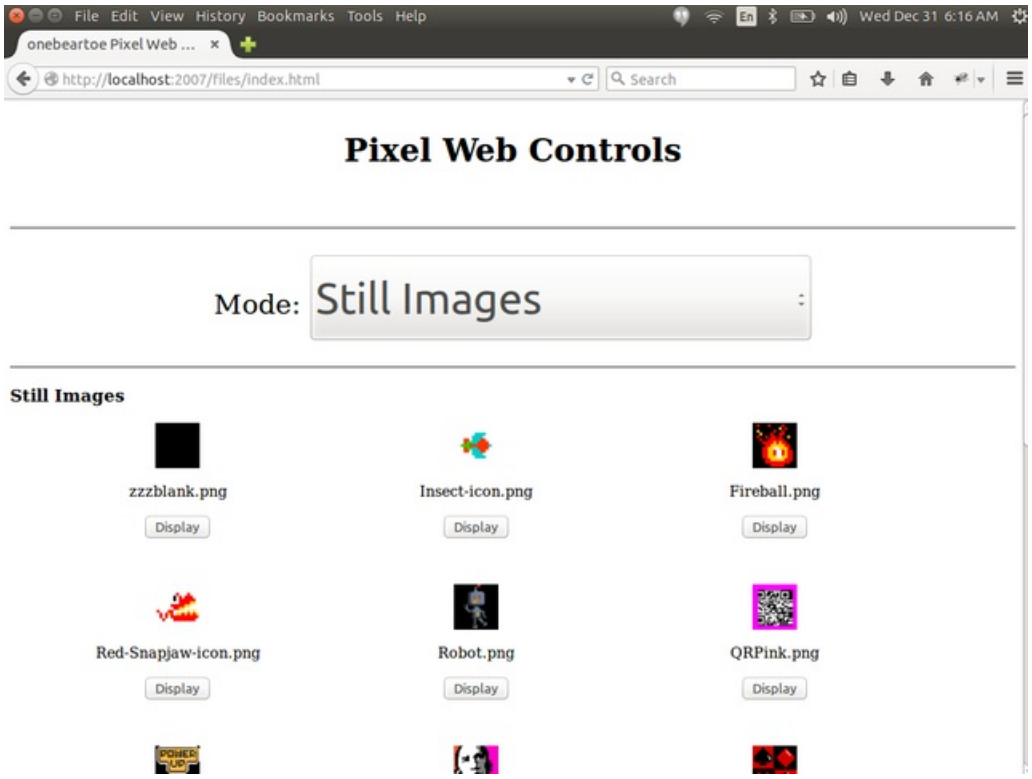


Displaying Stills and Animations

Still image mode presents the various images available for drawing on the Pixel.

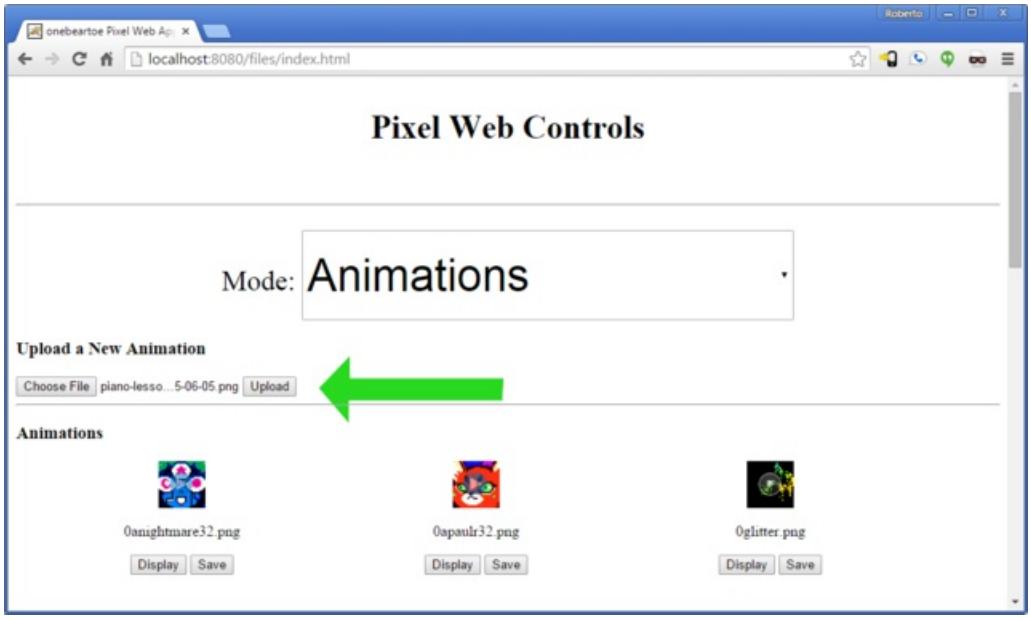
Animations mode similarly presents the various animations available.

On the user interface for still and animated images, a 'Display' button is used to select what is shown on the Pixel's RGB LED matrix.



Uploads for User Supplied Animated GIFs and Still Images

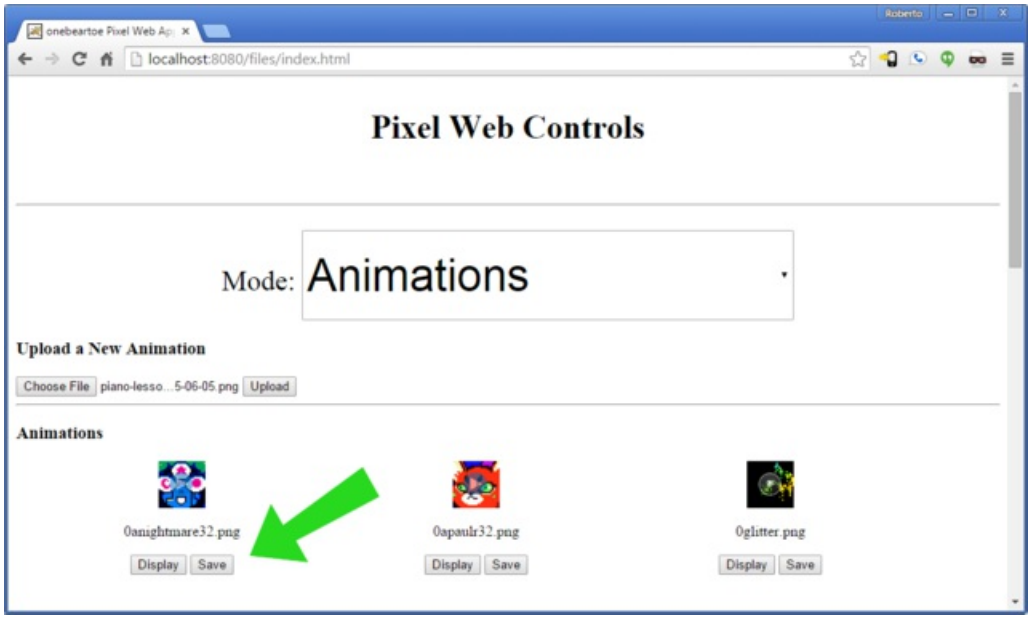
The user can upload their own still images and animated GIFs. The file chooser button is used to select a still or animated GIF, on the filesystem. Then click the upload button and it should appear in the corrosponding animated or still image list.



Save Animations to SD Card

In offline mode, an animation can be played from the SD card onboard the PIXEL. This means you can still have an animation play even if the Web enabled application is not running.

Use the 'Save' button to write an animation to the SD card for offline use.



Clock Mode

Clock mode presents an analog clock.

Command Line Switches

The applicaton has these command line swithces:

-h,--help	show help.
-l,--ledmatrix	Sets the LED matrix type. Default = 3 0=32x16 Seeed 1=32x16 Adafruit, 2=32x32 Seeed 3=PIXEL V2, 4=64x32 Seeed, 5=32x64 Seeed, 6=Seeed 2 Mirrored 7=Seeed 4 Mirrored (does not work), 8=128x32 Seeed, 9=32x128 Seeed 10=SUPER PIXEL 64x64, 11=32x32 Adafruit, 12=32x32 Adafruit Color Swap 13=64x32 Adafruit, 14=64x64 Adafruit, 15=128x32 Adafruit 16=32x128 Adafruit, 17=64x16 Adafruit The official list of supported LED panels is here: http://ledpixelart.com/pixelv2board/
-m,--matrix	Sets the LED matrix type, same as l option
-p,--port	Listening port. Default Port = 8080

Going Further

Create a Custom User Interface

A default user interface comes with the Web enabled Pixel app. It is available at:

http://rapberrypi:8080/files/index.html

A custom user interface to control Pixel can be made by posting to the following URLs, or Web API. Any programming language that can make HTTP requests can control the Pixel with this Web API.

Scrolling Text

http://rapberrypi:8080/text	change to scrolling text mode
http://rapberrypi:8080/text?t=some text to scroll	this will scroll 'some text to scroll' on the pixel
http://rapberrypi:8080/text/color/[hex-value] example: http://rapberrypi:8080/text/color/48301b	This sets the color of the scrolling text. The 'hex-value' given is used as the color.
http://rapberrypi:8080/text/speed/[delay-in-milliseconds] example: http://rapberrypi:8080/text/speed/300	This sets the scroll delay in milliseconds.

Still Images

http://rapberrypi:8080/still	Change to still images mode.
http://rapberrypi:8080/still/list	This URL returns a list of the available still images names. The image names are delimited by "-+-".
http://rapberrypi:8080/still/[image-name] examples: http://rapberrypi:8080/still/Robot.png http://rapberrypi:8080/still/Fireball.png	This writes the image denoted by the ID at the end of the URL. The ID is an integer that is used as an index into the still images available to the Web app.

Animations

http://rapberrypi:8080/animation	Change to animations mode.
http://rapberrypi:8080/animation/list	same as still images but for animations

http://rapberrypi:8080/animation/[animation-name] examples: http://rapberrypi:8080/animation/arrows.png http://rapberrypi:8080/animation/earth.png	same as still images but for animations
http://rapberrypi:8080/animation/save/[animation-name] examples: http://rapberrypi:8080/animation/save/arrows.png http://rapberrypi:8080/animation/save/earth.png	saves an animation to the SD card, for offline playback

Static Files

http://rapberrypi:8080/files/	static files are served from this URL context; HTML, CSS, JavaScript, image and animation previews
http://rapberrypi:8080/files/images/ examples: http://rapberrypi:8080/files/images/Fireball.png http://rapberrypi:8080/files/images/Robot.png	previews for the still images are under this URL context
http://rapberrypi:8080/files/animations/ examples: http://rapberrypi:8080/files/animations/arrows.png http://rapberrypi:8080/files/animations/earth.png	previews for the animations are under this URL context

Uploads

http://rapberrypi:8080/upload/	Uploads a HTTP post with a mulit-part file attachment. The name of the file parameter should be 'upload'. The request parameter named 'upload-type' is expected to have the value of 'STILL_IMAGE' or 'ANIMATED_GIF'.

Analog Clock

http://rapberrypi:8080/clock	Change to analog clock mode.

Modify the App

The source code for the Pixel application is on Github.

https://github.com/alinke/PIXEL (https://adafru.it/ejI)

The recomened way to make changes to the application is with the Netbeans IDE (https://adafru.it/ejJ). In Netbeans, open the directory where you cloned Pixel from Github. It will recognize and open a parent project labeled '**onebeartoe-pixel**'.

The '**onebeartoe-pixel**' project has other projects listed under '**Modules**'. Open the one labeled '**pixel-web-enabled**'. And make any change that better fit your setup. The F6 key compiles and runs the app locally.

Keep Up with Development Builds

More features are planned for the Web applicaiton for Pixel. Keep up with the development builds:

https://onebeartoe.ci.cloudbees.com/job/pixel/ (https://adafru.it/ejK)