

Required Hardware:

1. [ESP-EYE](#)
 - a. Optional [ESP-EYE case](#)
2. [PyPortal](#)
 - a. Optional [PyPortal case](#)
3. Two 3.3v usb to outler adapters and two usb to usb mini cables

OR

4. Two 3.3v micro usb wall outlet chargers

Build Steps:

1. Clone the [on-air](#) repo.

Cloudflare Worker:

1. Setup [Cloudflare](#) DNS records for your domain and endpoint, or setup a new [domain](#) with Cloudflare if you don't have one to resolve the endpoint.
2. Setup a [Cloudflare workers](#) account with worker K/V.
3. Setup the [Wrangler](#) CLI tool.
4. ``cd`` into the ``on-air/sighandler`` directory.
5. Update ``wrangler.toml``
6. Run ``wrangler preview``
7. ``wrangler publish``
8. Update ``Makefile`` with your domain and test calling.

PyPortal:

1. Setup CircuitPython 5.x on the [PyPortal](#).
 - a. If you're new to CircuitPython you should [read](#) this first.
2. Go to the directory where you cloned on-air.
3. ``cd`` into display.
4. Update ``secrets.py`` with your wifi information and status URL endpoint.
5. Copy ``code.py``, ``secrets.py`` and the bitmap files in ``screens/`` to the root of the PyPortal.
6. The display is now good to go.

ESP-EYE:

1. Setup ``esp-idf`` using the 4.1 release branch.
2. Install [espeak](#) and [sox](#).
3. Setup a Python 3.7 virtual environment and install Tensorflow 1.15.
4. ``cd`` into ``on-air/voice-assistant/train``
5. ``chmod +x orchestrate.sh`` and ``./orchestrate.sh``
6. Once training completes ``cd ../smalltalk``

7. Activate the `esp-idf` tooling so that `\$IDF_PATH` is set correctly and all requirements are met.
8. `idf.py menuconfig` and set your wifi settings.
9. Update the URL in `toggle_status.cc`
 - a. This should match the host and endpoint you deployed the Cloudflare worker to above
10. `idf.py build`
11. `idf.py --port <device port> flash monitor`
12. You should see the device start, attach to WiFi and begin listening for the wake word “visual” followed by “on” or “off”.