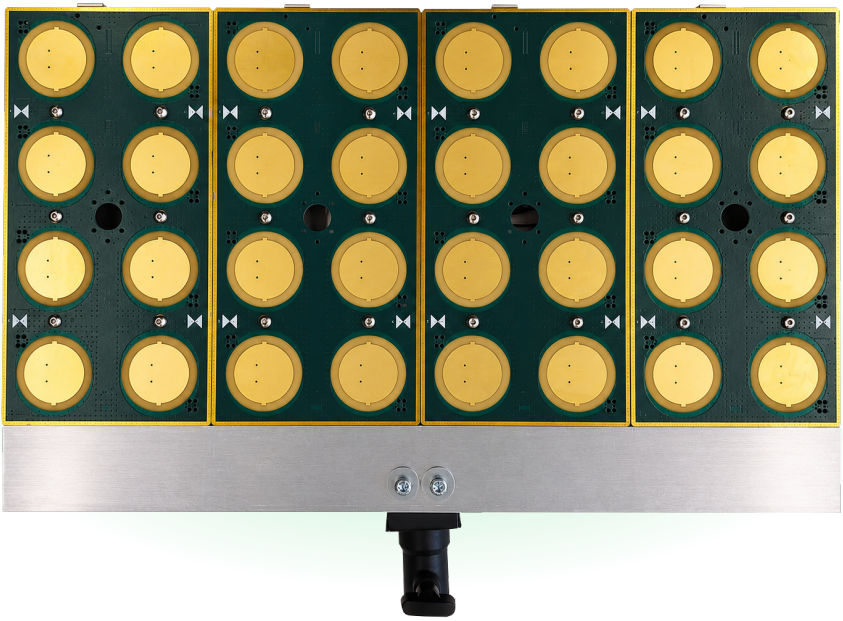
ESPARGOS



Aperture Kit 8 × 4

Four Phase-Coherent Arrays, One Large Aperture

Configuration Supplement · Rev 1.0, July 2026

ESPARGOS Aperture Kit 8 × 4

Configuration Supplement

Revision: 1.0

Date: July 2026

TTI GmbH / TGU ESPARGOS

Read together with

ESPARGOS One Quick Start Manual

Questions?

`info@espargos.net`

Note This supplement covers only what is *specific* to the pre-assembled **ESPARGOS Aperture Kit 8 × 4**: its factory default network and reference-signal settings, and the notes needed to get the combined array streaming. For everything about an individual array (mounting, power, web interface, `pyespargos`, synchronization principle) please refer to the **ESPARGOS One Quick Start Manual**.

Overview

The **ESPARGOS Aperture Kit 8 × 4** combines **four ESPARGOS One antenna arrays** into a single, larger phase-coherent aperture. The four boards are pre-mounted side by side and pre-wired for synchronization, so that their dual-polarized patch antennas act as one **8 × 4 array of 32 dual-polarized elements** at half-wavelength ($\lambda/2$) spacing. All four boards share a common 40 MHz clock and 2.4 GHz phase reference, so the Channel State Information (CSI) is captured coherently across the whole aperture, giving finer angular resolution than a single array.

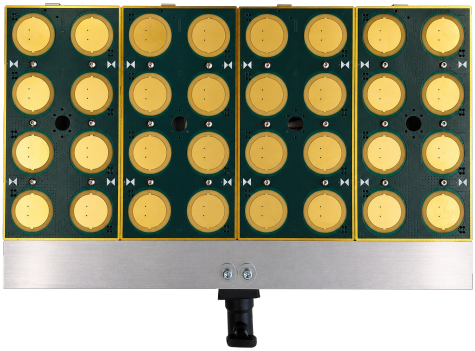


Figure 1. *The assembled **ESPARGOS Aperture Kit 8 × 4**: four **ESPARGOS One** arrays mounted side by side on a common bracket, forming one 8 × 4 aperture. From left to right the boards are* Array 1, Array 2, Array 3 *and* Array 4.

The four arrays are wired in a star topology through an **ESPARGOS Splitter** hidden inside the box: the third board from the left acts as the **master** that generates the shared reference, the other three are **slaves** that lock onto it. This wiring and the corresponding per-board settings are done at the factory; this supplement documents those defaults so they can be restored if a board is ever reset.

Note The four arrays connect to your network over Ethernet like four standalone units. Give them the fixed IP addresses listed in Section 4, because `pyespargos` needs to reach all four to combine their CSI.

Mounting

Seen from the back, the kit is one closed unit: the four sub-array enclosures sit on a common bracket, and apart from the Ethernet ports there is no external cabling. In particular, there are **no external synchronization cables**: the master's reference signal is amplified, split and distributed to all four boards entirely inside the enclosures, over the internal **ESPARGOS Splitter** and length-matched u.FL cables described in Section 3.

Mounting on a Tripod

Figure 2. *The spigot clamp on the mounting bracket.*

The mounting bracket carries a **spigot clamp** (Figure 2). To mount the kit on a tripod or light stand, insert the stand's **spigot** (the standard 16 mm stud used for studio lighting) into the clamp and tighten the locking knob. If you want to tilt the array, swivel mechanisms with a matching spigot are widely available; they are often sold as **umbrella adapters**.

Note To save space, the spigot clamp may be **shipped detached** from the bracket. In that case, attach it as shown in Figure 2 using the two supplied screws, washers and nuts.

Array Roles

Each of the four arrays has a fixed role in the synchronization chain. **Array 3** is the **master**: it generates the shared clock and phase reference and feeds it, via the ESPARGOS Splitter, to itself and to the other three boards. **Array 1**, **Array 2** and **Array 4** are **slaves** that lock onto that reference.

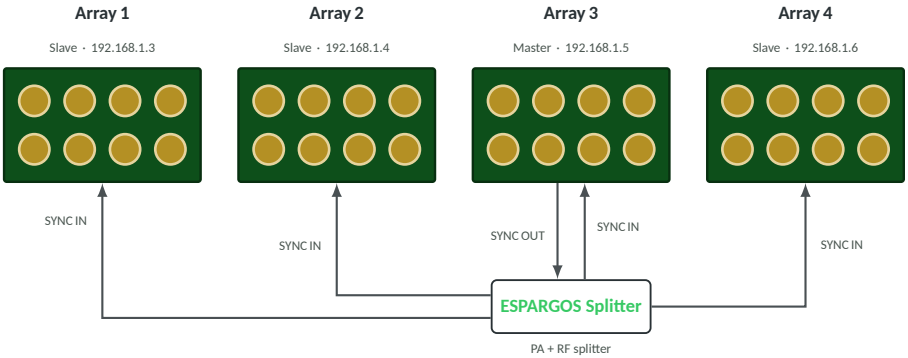


Figure 3. *Synchronization topology of the kit (SYNC cabling only; Ethernet omitted). The master (Array 3) drives the ESPARGOS Splitter, which returns the shared reference to all four boards over equal-length u.FL cables hidden inside the box. This wiring is pre-installed at the factory.*

Do Not Change the Reference Wiring The u.FL cables between the boards and the ESPARGOS Splitter are matched in length so that the four arrays stay phase-coherent. Do not swap, extend or shorten them, and do not change a board's **Reference Signal Source** unless you are deliberately reconfiguring the kit: a board set to *Slave* with no signal on SYNC IN will fail to boot.

Factory Default Settings

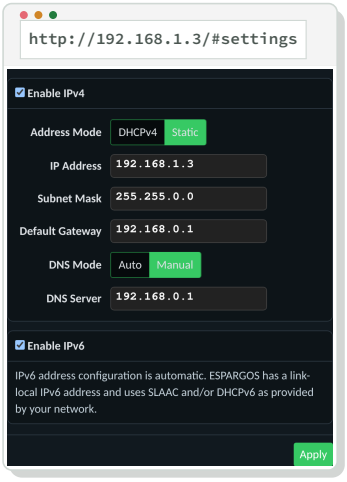
The table below summarizes the factory configuration of the four arrays. Each board is reached under its own fixed IPv4 address; the reference-signal role is set on the **Reference Signal and Channel** card of that board's web interface.

Setting	Array 1	Array 2	Array 3	Array 4
IP address	192.168.1.3	192.168.1.4	192.168.1.5	192.168.1.6
Role	Slave	Slave	**Master**	Slave
Ref. Signal Source	Slave (IN only)	Slave (IN only)	Master (OUT + IN)	Slave (IN only)
Generate Phase Ref.	Never	Never	During Calibration	Never

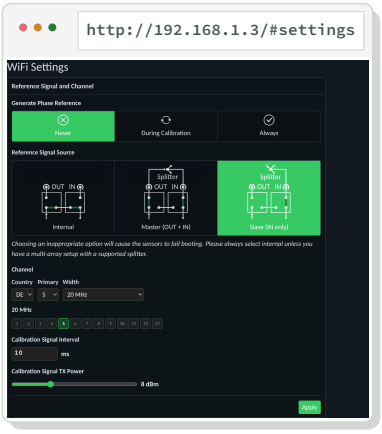
Common to all four boards: subnet mask `255.255.0.0`, default gateway and DNS server `192.168.0.1`, calibration signal interval 10 ms and calibration TX power 8 dBm. Adjust the network addresses and the gateway to suit your own network if needed.

Keep WiFi Settings Consistent The WiFi settings, in particular the primary and secondary channel, must be **identical on all four sub-arrays**; the kit can only combine CSI if all boards receive on the same channel. `pyespargos` takes care of this automatically: it applies the channel configured in its pool settings to every board it connects to. If you change WiFi settings manually in the web interfaces, apply the same values to all four boards.

Array 1: Slave (192.168.1.3)



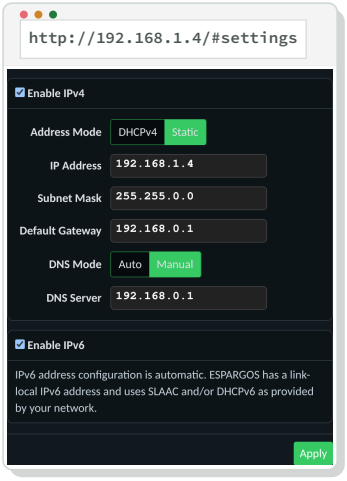
(a) *Network settings*



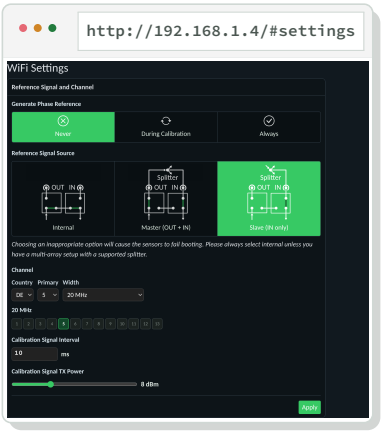
(b) *Reference signal & channel*

Figure 4. *Array 1 default settings: static IP `192.168.1.3`; reference signal source **Slave**, phase reference **Never**.*

Array 2: Slave (192.168.1.4)



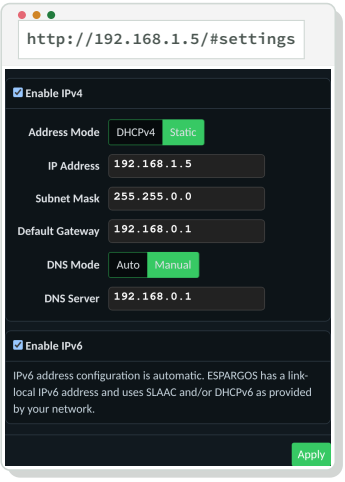
(a) *Network settings*



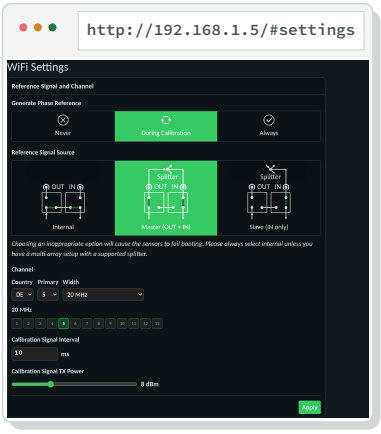
(b) *Reference signal & channel*

Figure 5. *Array 2 default settings: static IP `192.168.1.4`; reference signal source **Slave**, phase reference **Never**.*

Array 3: Master (192.168.1.5)



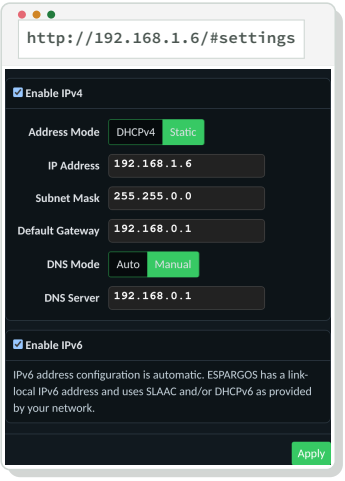
(a) *Network settings*



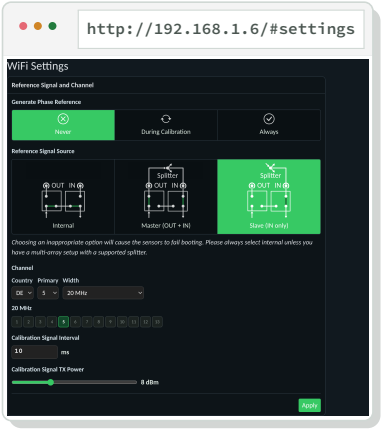
(b) *Reference signal & channel*

Figure 6. *Array 3 default settings: static IP* `192.168.1.5`*; reference signal source* ***Master****, phase reference* ***During Calibration****. This is the only board that generates the shared reference.*

Array 4: Slave (192.168.1.6)



(a) *Network settings*



(b) *Reference signal & channel*

Figure 7. *Array 4 default settings: static IP* `192.168.1.6`*; reference signal source* ***Slave****, phase reference* ***Never****.*

Bringing Up the Kit

1. **Check each array in the browser.** Open the web interface of each board in turn (`192.168.1.3`, `.4`, `.5`, `.6`) and confirm it is reachable. On each **Home** page, verify that all antennas of the sub-array receive a sensible signal.
2. **Confirm the WiFi settings match.** Make sure the WiFi settings (primary and secondary channel) are identical on all four arrays. `pyespargos` keeps them consistent automatically, but a channel mismatch after manual changes is the most common cause of a kit that will not combine.
3. **Stream the raw CSI** from all four boards at once with `pyespargos`:

```
./demos/instantaneous-csi/instantaneous-csi.py
     192.168.1.3,192.168.1.4,192.168.1.5,192.168.1.6
```

4. **Configure the combined array and run a demo** once the raw CSI looks good: point any full-aperture demo at the combined-array configuration file (Section 6), for example the camera overlay or the 3D radiation pattern (Section 7).

Note If a sub-array behaves strangely, a **reboot** often helps: use the reboot button on the **Home** page of that board's web interface.

Notes on WiFi Measurements

- WiFi signals can behave unintuitively: transmitter antennas are often directional, and objects in the room reflect signals strongly, so measured signals may arrive from unexpected directions.
- It is worth taking the time to get familiar with the many display and processing options offered by the graphical interface.

Combined-Array Configuration

Every `pyespargos` demo that treats the kit as a single 8 × 4 aperture (the camera overlay, the 3D radiation pattern, the `combined-array` phase viewer and others) needs to know which boards make up the array and how their antennas are arranged on the grid. This is described *once* in a **combined-array configuration file** and passed to any such demo with the `-c` option. `pyespargos` ships with a ready-made config file, `config/aperture-kit-8x4.yml`:

```
combined-array:
  boards:
    ba1:
      host: 192.168.1.3
      cable:
        length: 0.0
        velocity_factor: 0.76
    ba2:
      host: 192.168.1.4
      cable:
        length: 0.0
        velocity_factor: 0.76
    ba3:
      host: 192.168.1.5
      cable:
        length: 0.0
        velocity_factor: 0.76
    ba4:
      host: 192.168.1.6
      cable:
        length: 0.0
        velocity_factor: 0.76
  array:
    - - "ba1.1.0"
      - "ba1.0.0"
      - "ba2.1.0"
      - "ba2.0.0"
      - "ba3.1.0"
      - "ba3.0.0"
      - "ba4.1.0"
      - "ba4.0.0"
    - - "ba1.1.1"
      - "ba1.0.1"
      - "ba2.1.1"
      - "ba2.0.1"
      - "ba3.1.1"
      - "ba3.0.1"
      - "ba4.1.1"
      - "ba4.0.1"
    - - "ba1.1.2"
      - "ba1.0.2"
      - "ba2.1.2"
      - "ba2.0.2"
      - "ba3.1.2"
      - "ba3.0.2"
      - "ba4.1.2"
      - "ba4.0.2"
    - - "ba1.1.3"
      - "ba1.0.3"
      - "ba2.1.3"
      - "ba2.0.3"
      - "ba3.1.3"
      - "ba3.0.3"
      - "ba4.1.3"
      - "ba4.0.3"
```

Note Each entry `baN.row.col` places antenna (`row`, `col`) from board `baN`'s 2 × 4 addressing grid (row indices `0–1`, column indices `0–3`) at that position of the combined array. All eight antennas of each board must appear exactly once and form one contiguous block, rotated only in steps of 90°: here, each board fills a rotated 4 × 2 block. If you change the boards' IP addresses, update the `host` fields.

The file describes *only* the array geometry: the four hosts and their antenna mapping. Because that geometry is fixed for the kit, the *same* file is reused by every combined-array demo (Section 7).

WiFi, Channel and RF Settings

Runtime receiver settings such as the WiFi channel, the RF switch position or the CSI acquisition mode are **not** part of the combined-array file. They describe how the boards receive, not how they are arranged, and are best chosen per measurement. Supply them on the command line with one or more `-o KEY=VALUE` options, which override the built-in defaults. The line below is only an **example** (reasonable starting values for the camera demo); choose values that suit your own setup:

```
-o pool.channel=5 -o pool.secondary_channel=0 -o pool.rf_switch=2 -o
     pool.acquire_lltf_force=true
```

- `pool.channel`: primary WiFi channel (1-13) the boards tune to. Pick the channel your traffic of interest uses.
- `pool.secondary_channel`: secondary channel for 40 MHz (HT40) capture. `0` keeps a single 20 MHz channel; other values place the secondary channel above or below the primary.
- `pool.rf_switch`: which antenna feed the on-board RF switch selects. `2` and `3` pick the two orthogonal polarizations, `4` alternates between them.
- `pool.acquire_lltf_force`: when `true`, always capture the legacy L-LTF CSI (20 MHz) regardless of packet format, giving a uniform CSI format across mixed traffic.

These options only set the *initial* values. Every demo also exposes the same settings as live **Receiver Settings** controls in its window, so you can change the channel, RF switch and the rest while the demo is running, without restarting it or editing any file.

Note `pyespargos` applies the `pool.channel` (and secondary channel) it receives to *all four* boards, so they always listen on the same channel (see the warning on page 6).

Running the Demos

With the configuration file from Section 6 in place, every full-aperture demo is launched the same way: point it at **`config/aperture-kit-8x4.yml`** with **`-c`**, and add any receiver overrides (Section 6.1) with **`-o`**.

Camera Overlay

For the camera-overlay demo, mount a **webcam above the array** looking out over the same scene, then start:

```
./demos/camera/camera.py -c config/aperture-kit-8x4.yml \
    -o pool.channel=5 -o pool.secondary_channel=0 \
    -o pool.rf_switch=2 -o pool.acquire_lltf_force=true
```

The demo estimates the direction the received WiFi signals arrive from and overlays that angular spectrum onto the live camera image.

Other Combined-Array Demos

The same configuration file drives the other full-aperture demos; only the script changes (and the **`-o`** overrides, as needed). For example, the **3D radiation pattern** demo renders the aperture's beam pattern in an interactive 3D scene:

```
./demos/radiation-pattern-3d/radiation-pattern-3d.py -c config/aperture-kit-8x4.yml
```

and the **`combined-array`** demo shows the average received phase of every antenna across the whole 8 × 4 grid:

```
./demos/combined-array/combined-array.py -c config/aperture-kit-8x4.yml
```

> **Note** The 3D radiation pattern demo needs a few extra Python and system packages; see **`demos/radiation-pattern-3d/README.md`** in **`pyespargos`**. Refer to the **`pyespargos`** documentation for the full list of demos that accept a combined-array configuration.

Notes

Questions?

info@espargos.net

https://espargos.net