My Sanbot — desktop remote & Nano dependencies

Quick setup for the desktop remote (Windows / Mac / Linux) and the Sanbot Nano packages it needs on the robot.

Download page: https://brien.com.au/apps/sanbot.html
Desktop packages (My Sanbot Web 1.0.10, self-contained — no .NET install):
`mysanbot-web-win-x64.zip` · `mysanbot-web-osx-arm64.zip` · `mysanbot-web-osx-x64.zip` · `mysanbot-web-linux-x64.zip`
Nano pack (optional install kit): `mysanbot.zip`

1. Desktop (Windows / Mac / Linux)

1. Download the zip for your OS from the download page.
2. **Windows:** unzip and run `MySanbot.Web.exe`.
   Mac / Linux: unzip, then `chmod +x MySanbot.Web && ./MySanbot.Web`. On Mac, if Gatekeeper blocks: `xattr -dr com.apple.quarantine MySanbot.Web`.
3. A browser opens `http://127.0.0.1:5177/`.
4. In **Configuration**, set **Robot host** to the Nano Wi-Fi IP.
5. Prefer transport **OpenSDK WS** (port **9181**) for control and IR radar. Or **Sanbot Controller (HTTP+SSE)** on Proxy HTTP **9182**.
6. **Show IP addresses** is off by default — masks Host, status, API URLs, and Logs (e.g. `192.168.1**.***`). Turn it on in Configuration when you need the full address.

The PC and Nano must be on the same LAN (or VPN). Allow inbound TCP **5177** on the PC only if phones on the LAN will open this UI.

2. Nano — required (control + radar)

File	Package	Notes
`MySanbotOpenSdkProxy-*-debug.apk`	`mysanbot.opensdkproxy`	OpenSDK Proxy **1.3.79+** — default WS **:9181**, HTTP/SSE **:9182** (Listen ports configurable on-device). Keep running with MainService **bound** for IR/PIR radar. Forehead MJPEG needs 1.3.79+.

3. Nano — optional

File	Package	For
`Sanbot_Ros_Bridge-*-ws-debug.apk`	`com.grin.sanbotrosbridge`	Body / forehead MediaMTX path, Direct WS :8080
`SanbotHumanFollow-debug.apk`	`com.qihan.HumanFollow`	Follow mode

My Sanbot for Android (`mysanbot-android.apk`, package `mysanbot.app`) is the phone remote — install it on an Android phone or tablet, **never on the Nano**. It is not in the robot pack: on the Nano, `adb install` fails with `INSTALL_FAILED_NO_MATCHING_ABIS` because the robot is 32-bit Android 6.

4. Where the robot APKs live

- Website Nano pack: `mysanbot.zip` → `apk/` (same page as above)
- Local develop folder: `~/Developer/SanBot Nano/deploy-pack-20261027/apk/`

5. Updating robot apps later

You don't need the pack again for updates. In the remote, **Software** → **Check for Updates** lists newer builds of the Proxy, ROS Bridge, Human Follow and other robot apps, plus published apps the robot doesn't have yet such as Vitalia (**Install**), and switches to the Updates filter when it finds any (also on the Proxy page, `http://<robot-ip>:9182/` → Software). **Update** makes the robot download and check the app, then opens Android's installer on the robot's screen.

1. **Tap Install on the robot's screen.** Unless the robot is rooted, the remote can't approve installs for you.
2. **Not next to the robot?** Mirror its screen on your computer with scrcpy and click Install there: `scrcpy --serial <robot-ip>:5555 --no-audio`. In My Sanbot Web it's **Software** → **Open robot screen**. Needs Wi-Fi ADB on; the first time, allow debugging from that computer on the robot. Full steps: `docs/Scrcpy_Remote_Desktop.md` in the robot pack.
3. **Wait.** The robot shows *Installing…* for up to a few minutes while Android prepares the app (about 3 minutes for ROS Bridge). Then tap **Done**.
4. **Start the app again** with Launch in Software. Updating the Proxy itself drops the connection until it restarts — Connect again.

Install blocked on the robot means Unknown sources is off. Tap **Settings** in that dialog and turn it on (the pack's install script does this), or run `adb shell settings put secure install_non_market_apps 1`, then press Update again.

Radar reminder: High-rate IR needs OpenSDK Proxy with MainService bound, plus **Sensors On** in the remote. Battery/status can still arrive when unbound; `/ir` will not. Older Proxy builds that still listen on 8081/8082 are healed in Config, but prefer matching the on-device Listen ports.
