

HANDS-ON WORKSHOP

Build your own ADS-B receiver and put your sky on the map

Everything in this course can be checked from your own window. This workshop tells you exactly what to buy, how to put it together, how to install the Dataero feeder on a Raspberry Pi and how to see the aircraft you hear appear on radar.dataero.eu, credited to your station. No radio, electronics or Linux experience is needed. Budget about 100 to 150 euros, one afternoon, and a window with a view of the sky.

Learn - <https://radar.dataero.eu/learning/workshop> 2026-10-22

CONTENTS

BEFORE YOU START What you will build, and why it is worth it

ONE PICTURE How the parts fit together

SHOPPING LIST What to buy, part by part

WHERE IT GOES Placing the antenna: height, sky and cable

STEP 1 Assembling the hardware

STEP 2 Preparing the Raspberry Pi

STEP 3 Installing the Dataero feeder

STEP 4 Going live, and claiming your receiver

STEP 5 Tuning, and understanding what you see

WHEN IT DOES NOT WORK Troubleshooting

AFTERWARDS Keeping it running

FOR A CLASS OR A FAMILY Turning the receiver into a project

WELCOME What feeders get, and what the map gets from you

BEFORE YOU START

What you will build, and why it is worth it

A receiver is four things in a line: an antenna that catches the 1090 MHz wave, a USB dongle that turns the wave into numbers, a small computer that decodes the numbers into messages, and an internet connection that carries the decoded positions to Dataero. From the moment the feeder is running, every aircraft your antenna hears is drawn on the live map at **radar.dataero.eu**, for everyone, with your station named as the source. Nothing in the chain transmits: you only listen, and listening is legal in Belgium, across the European Union and in most of the world (if you live elsewhere, check the rules on receiving equipment in your country).

The course explained why the system works; the receiver lets you test it. You will measure the radio horizon of chapter 7 against real aircraft, see the inverse square law in the signal strength column, watch the message rate change with the time of day, and compare a window antenna with a roof antenna. For a family it is the project that ends the story; for a class it is a permanent instrument on the roof that produces real data all year.

It also makes you a **feeder**. Dataero's map is built entirely from receivers hosted by volunteers, airports and schools, and the map is only as good as its listeners. A feeder gets, free, what a visitor does not: a dashboard of what the station hears, a coverage map of its patch of sky in two and three dimensions, 365 days of flight history, and a place on the leaderboard against every other station on the network. Nothing is charged, nothing is signed, and the same receiver can keep feeding any other network you already use.

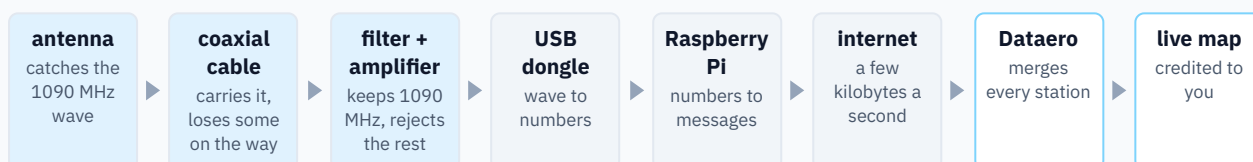
Time, money, skills

Money: 150 to 200 euros for the basic build, bought new, 300 to 420 for the advanced build; less if you already own a Raspberry Pi. **Time:** one afternoon once the parts have arrived - about 30 minutes of assembly, 10 minutes to prepare the memory card, 10 minutes to install the feeder.

Skills: screwing a connector, copying three lines into a terminal. **Running cost:** the receiver draws 4 to 5 watts, about 40 kWh a year, roughly 10 to 15 euros of electricity.

ONE PICTURE

How the parts fit together



Signal flows left to right. The three stages on the left are radio, the next two are computing, the last three are the network. Each stage has a chapter in the reference course.

The antenna is where the physics of chapters 5 to 8 happens: a quarter-wave of metal, about 65 mm long, resonates with the 27.5 cm wave and turns a few billionths of a watt into a tiny voltage. The cable carries that voltage to the dongle and loses roughly half of it every ten metres if it is thin, which is why the antenna should be close to the dongle or the cable should be thick. The filter and amplifier are optional: they strengthen the 1090 MHz signal before it reaches the dongle and reject the mobile-phone and FM-radio transmitters that would otherwise deafen it. The dongle is a software-defined radio that samples the wave two million times a second and hands the numbers to the Raspberry Pi, where the decoder (readsb, a descendant of dump1090) finds the preamble of chapter 4, reads the 112 bits, checks the error code and discards anything that fails. The Dataero feeder then forwards the decoded messages over your internet connection to Dataero's ingest hub, which merges your station with every other one into the map.

SHOPPING LIST

What to buy, part by part

The list below is complete: with these parts and nothing else you will be feeding. It comes in two builds. The basic build keeps a small antenna indoors, at a window or in the loft, and is enough to see the system work and to feed from the first evening; the advanced build puts a tuned antenna outdoors, with a filtered amplifier and a low-loss cable, and reaches the radio horizon. Prices are indicative, new, including VAT, and vary by shop and by month. The Buy buttons open each part on the Amazon store of the country you choose; the same parts are sold by any electronics shop, and the computer by the official

Raspberry Pi resellers (the Raspberry Pi website lists them by country). Whatever the shop, buy the dongle sold for ADS-B, because the same dongle sold as a television tuner is often an older, noisier version.

The computer: a Raspberry Pi

A **Raspberry Pi 4 Model B with 2 GB** of memory is the right choice: it decodes 1090 MHz with a fraction of its power, it has four USB ports, wired Ethernet and Wi-Fi, and it is sold with a case and power supply in a kit. A Raspberry Pi 5 works just as well and costs a little more; a Raspberry Pi 3B+ works if you already have one; a Raspberry Pi Zero 2 W works but is tight on USB power and best avoided for a first build. Buy the **official power supply** for your model (a weak phone charger is the most common cause of a receiver that reboots at random), a **microSD card of 16 or 32 GB** from a known brand, and a **case** with some ventilation, because the dongle and the processor both get warm. Add the small heatsink set if the kit does not include one. Expect 50 to 70 euros for the computer, 10 to 15 for the power supply, 10 for the card and 10 for the case.

The radio: an RTL-SDR dongle

The dongle is a software-defined radio built around an RTL2832U chip and an R820T2 or R860 tuner, the same family that chapter 9 described. Buy one **sold for ADS-B**. The best starting choice is a dongle with a **built-in 1090 MHz filter and amplifier** (the FlightAware Pro Stick Plus is the best known; AirNav's FlightStick is similar): the filter and the low-noise amplifier that you would otherwise buy separately are inside the stick, and it hears two to three times further than a plain dongle in a city. A general-purpose dongle (the RTL-SDR Blog V4, a Nooelec NESDR SMARTEE) also works and is more versatile if you want to listen to other things later, but add the external filter described below if you live within a few kilometres of mobile-phone masts. Whichever you buy, check the antenna connector: **SMA** is the standard and every antenna and cable in this list uses it; avoid dongles with the small MCX connector, which needs an adapter and breaks. Expect 30 to 45 euros.

The antenna

Start with the small magnetic antenna that comes in the dongle box, if there is one: on a metal surface at a window it hears aircraft 80 to 150 km away and is enough to see the system work on the first evening. The real receiver, though, needs a **tuned 1090 MHz antenna**, and this is the part not to save on: a stacked collinear in a weatherproof tube, 60 to 70 cm long, with 4 to 6 dBi of gain and an N or SMA connector at the bottom (FlightAware's 1090 MHz antenna and Jetvision's A3 are the usual choices, 50 to 70 euros). It is built to resonate at exactly the wavelength of chapter 8 and to concentrate its sensitivity towards the horizon, where the distant aircraft are; outdoors, on a short mast with a clear view, it reaches the radio horizon, 350 to 450 km for aircraft at cruise altitude. Home-made antennas cut from wire are a classroom experiment, not a receiver: a few millimetres of error detunes them and they hear a fraction of what a factory-tuned antenna hears.

Cable and connectors

If the antenna is next to the Pi, a **1 to 3 m SMA male to SMA male cable** of a thin, flexible type (RG-174 for a metre, RG-58 or H155 for three) is enough. The longer the run, the thicker the cable must be: at 1090 MHz a thin cable loses about 3 dB, half the signal, every ten metres, whereas a thick low-loss cable

such as LMR-400 or H1000 loses about 1.3 dB. A roof antenna 15 m from the Pi therefore wants LMR-400 or a filtered amplifier at the antenna, or both. Buy the cable made to length with the connectors fitted (N male at the antenna end if the antenna has an N socket, SMA male at the dongle end), because fitting an N connector by hand needs a crimping tool and practice. Expect 10 euros for a short cable, 30 to 60 for a long low-loss one. Add a **short USB extension cable**, 30 to 50 cm, to move the dongle away from the Pi: a dongle plugged straight into the board picks up the processor's own radio noise and blocks the neighbouring port.

Filter and amplifier (optional, but the cheapest range you can buy)

If your dongle has no built-in filter, a **1090 MHz band-pass filter** (20 to 25 euros, SMA in and out) between the cable and the dongle removes the mobile-phone, FM, television and pager signals that a plain dongle also receives and that overload it near a city. Better still is a **filtered low-noise amplifier** (about 35 euros; the RTL-SDR Blog 1090 LNA and the Uputronics preamplifier are the common ones), mounted at the antenna end of a long cable: it amplifies the signal before the cable loses it, and it is powered by the dongle through the cable itself (a *bias tee*, which the dongle must support; the Blog V4, the Pro Stick Plus and the SMARtee all do). An amplifier at the Pi end of a long cable is almost useless, because it amplifies what the cable has already lost.

Tools and the rest

You need a computer with an SD card reader (or a USB card reader, 5 euros) to write the memory card, your home router's Wi-Fi password or a free Ethernet port and cable, and, for the antenna, a window ledge, a loft beam or a short mast with two wall brackets. For an outdoor antenna add self-amalgamating tape to seal the connector (3 euros), cable ties and a few metres of cable slack for a drip loop. A small screwdriver and scissors are the only tools.

Basic build Small antenna indoors, at a window or in the loft - 50 to 150 km - 150 to 200 EUR

Part	What it is for	Buy on Amazon
Raspberry Pi 4 kit Model B with 2 or 4 GB, the official power supply, a ventilated case and a 32 or 64 GB microSD card, all in the kit; a Raspberry Pi 5 kit works just as well	Decodes the messages and runs the Dataero feeder	https://radar.dataero.eu/go/pi
AirNav Radar FlightStick dongle USB software-defined radio with a built-in 1090 MHz filter and amplifier, SMA socket; the FlightAware Pro Stick Plus, while stocks last, is equivalent	Turns the radio wave into numbers; the built-in filter and amplifier are what give it range in a city	https://radar.dataero.eu/go/sdr
USB extension cable, 30 to 50 cm USB 2.0, A male to A female	Moves the dongle away from the processor's own radio noise and frees the neighbouring port	https://radar.dataero.eu/go/usb-ext

Part	What it is for	Buy on Amazon
Small 1090 MHz antenna with a magnetic base Tuned for ADS-B, with its own cable and an SMA male plug	Catches the wave at a window; on a metal surface it hears aircraft 50 to 150 km away	https://radar.dataero.eu/go/antenna-indoor

Advanced build

Outdoor antenna on a short mast, for long range - 250 to 400 km - 300 to 420 EUR

Part	What it is for	Buy on Amazon
Raspberry Pi 4 kit Model B with 2 or 4 GB, the official power supply, a ventilated case and a 32 or 64 GB microSD card, all in the kit; a Raspberry Pi 5 kit works just as well	Decodes the messages and runs the Dataero feeder	https://radar.dataero.eu/go/pi
AirNav Radar FlightStick dongle USB software-defined radio with a built-in 1090 MHz filter and amplifier, SMA socket; the FlightAware Pro Stick Plus, while stocks last, is equivalent	Turns the radio wave into numbers; the built-in filter and amplifier are what give it range in a city	https://radar.dataero.eu/go/sdr
USB extension cable, 30 to 50 cm USB 2.0, A male to A female	Moves the dongle away from the processor's own radio noise and frees the neighbouring port	https://radar.dataero.eu/go/usb-ext
FlightAware 1090 MHz outdoor antenna Collinear in a weatherproof tube, 66 cm, N female socket; Jetvision's A3 is the alternative	Built to resonate at exactly 27.5 cm and to look at the horizon: outdoors, with a clear view, it reaches the radio horizon	https://radar.dataero.eu/go/antenna-outdoor
Filtered 1090 MHz amplifier RTL-SDR Blog ADS-B LNA, Nooelec SAWbird+ ADS-B or the Uputronics preamplifier, powered through the cable (bias tee)	Mounted at the antenna, it strengthens the signal before the cable loses it	https://radar.dataero.eu/go/lna
N male to SMA male adapter Between the antenna's N socket and the amplifier's SMA socket	Joins the antenna to the amplifier	https://radar.dataero.eu/go/adapter
Low-loss coaxial cable, 10 m LMR-400 or H1000, SMA male at both ends, made to length with the connectors fitted	Carries the signal from the roof to the dongle and loses only about 1.3 dB per 10 m	https://radar.dataero.eu/go/coax

Part	What it is for	Buy on Amazon
Short mast with wall brackets About 1 m long, 30 to 40 mm in diameter, with two brackets	Lifts the antenna above the roof line, clear of walls and chimneys	https://radar.dataero.eu/go/mount
Self-amalgamating tape One roll	Seals every outdoor connector against rain	https://radar.dataero.eu/go/tape

As an Amazon Associate, Dataero Europe earns from qualifying purchases made through these links, at no extra cost to you. Prices, stock and delivery are those shown by Amazon; the same parts can be bought from any electronics shop.

What not to buy

A 10-euro television stick with an unnamed chip and an MCX connector (it works for a week, drifts, overheats and loses the weak messages); a 'wideband' discone or a car-radio antenna (not resonant at 1090 MHz; chapter 8 explains why length matters); a Wi-Fi or 4G antenna with the right-looking connector (wrong frequency); a long cable with crimp-it-yourself connectors; a phone charger as the power supply; a used Raspberry Pi of unknown origin with a card already in it.

WHERE IT GOES

Placing the antenna: height, sky and cable

Placement beats price. The radio horizon formula from chapter 7, $d = 4.12 \times (\sqrt{h_{\text{aircraft}}} + \sqrt{h_{\text{antenna}}})$ with heights in metres, says that an antenna at 2 m on a window ledge sees an airliner at 11 000 m up to 438 km away, and the same antenna on a 20 m roof only 13 km further. Height matters far less than what is **in the way**: a wall, a hill, a metal roof or a neighbouring building blocks the line of sight completely, and a window facing away from the airway hears half the sky. So the first question is not 'how high' but 'what can the antenna see', and the answer should be: as much sky as possible, down to the horizon, in as many directions as possible.

At a window, put the antenna on the outer side of the glass if you can (on a balcony rail, held by its magnet on a steel plate), otherwise on the inner ledge against the glass; modern double glazing with a metallic coating weakens the signal noticeably, and a window that opens lets the cable out. Choose the window that faces the nearest airport or the busiest airway; the live map tells you which. **In a loft**, a roof of tiles or slates is almost transparent to 1090 MHz and the antenna sees every direction; a metal roof or foil-backed insulation is opaque. Hang the antenna from a beam, as high as the ridge allows, away from water tanks and wiring. **On a roof**, a mast of a metre above the ridge on two wall brackets clears the chimney and the neighbours; this is the placement that reaches the radio horizon, and the one that needs a thick cable or an amplifier at the antenna.

Cable	Loss per 10 m at 1090 MHz	Signal left after 10 m	Use
RG-174 (2.8 mm)	about 7 dB	20 %	inside, under 1 m
RG-58 (5 mm)	about 5 dB	30 %	inside, 1 to 3 m
H155 / LMR-200 (5.5 mm)	about 3 dB	50 %	up to 5 m
LMR-400 / H1000 (10 mm)	about 1.3 dB	75 %	roof runs of 10 to 20 m

Cable	Loss per 10 m at 1090 MHz	Signal left after 10 m	Use
amplifier at the antenna + any cable	loss paid before the cable	nearly all	any long run

Safety

Never put an antenna where it could touch or fall on an overhead power line; keep it below the roof's highest point if the building has a lightning conductor, and otherwise accept that a small plastic antenna on a 1 m mast does not attract lightning more than the chimney does, but unplug the dongle in a thunderstorm. Roof work is for an adult, in daylight, with someone holding the ladder; a loft or a window gives 80 % of the result with none of the risk. Seal every outdoor connector with self-amalgamating tape and leave a loop in the cable below the entry point so that rain drips off instead of running in. The receiver only listens: nothing here radiates.

STEP 1

Assembling the hardware

Assembly is screwing connectors together; the only thing to get right is to tighten each one by hand until it stops turning, with no tool, and to avoid bending the cable sharply within a hand's width of a connector.

1. Fit the heatsinks to the Pi if the kit has them, put the Pi in its case and leave the lid off for now.
2. Screw the antenna cable onto the antenna (N or SMA, hand-tight), and if you use a separate filter or amplifier, screw it onto the dongle end of the cable: antenna, cable, filter or amplifier, dongle, in that order. An amplifier at the antenna goes between the antenna and the cable instead.
3. Screw the cable onto the dongle's SMA socket. Plug the dongle into the USB extension cable and the extension into one of the Pi's blue USB 3 ports (any port works; the blue ones are furthest from the Wi-Fi antenna).
4. Place the antenna where you decided above and route the cable with gentle curves. Put the dongle somewhere it can get air; it runs warm to the touch, which is normal.
5. Connect the Ethernet cable to the router if you use one. Do not plug the power in yet: the memory card comes first.

STEP 2

Preparing the Raspberry Pi

The Pi needs an operating system on its memory card and a way for you to type commands into it from your own computer. Both take ten minutes and are done by clicking in one program, **Raspberry Pi Imager**, which you download free from the Raspberry Pi website for Windows, macOS or Linux.

1. Put the microSD card in your computer. Open Raspberry Pi Imager. Under 'Raspberry Pi Device' choose your model; under 'Operating System' choose **Raspberry Pi OS (other)**, then **Raspberry Pi OS Lite (64-bit)**, the version without a desktop, because the receiver never has a screen; under 'Storage' choose the card.
2. Click Next and, when Imager offers to apply OS customisation settings, click **Edit settings**. On the General tab set the hostname to **adsb** (so the Pi answers to adsb.local), tick 'Set username and

password' and choose both (write them down), and enter your Wi-Fi name and password with your country unless you use an Ethernet cable; set your time zone. On the Services tab tick **Enable SSH** with password authentication. Save, then confirm that the card will be erased and let Imager write and verify it.

- Put the card in the Pi, plug in the power supply and wait two minutes: the green light flickers while it starts, and the first start is slower.
- On your computer open a terminal: on Windows, PowerShell (search for it in the Start menu); on a Mac, Terminal; on Linux, any terminal. Type the command below with the username you chose, press Enter, answer **yes** to the question about the fingerprint, then type your password (nothing shows while you type) and press Enter. The prompt changes to the Pi's: you are now typing on the receiver.

```
ssh <username>@adsb.local
```

If *adsb.local* is not found, your router does not pass local names: open its web page, find the list of connected devices, note the Pi's IP address (something like 192.168.1.42) and use that instead of the name. Once connected, bring the system up to date and install git, the tool that downloads the feeder; copy the line, paste it, press Enter and wait. The update can take ten minutes the first time.

```
sudo apt update && sudo apt full-upgrade -y && sudo apt install -y git
```

STEP 3

Installing the Dataero feeder

The Dataero feeder is a small, open-source program published by Dataero on GitHub (github.com/dataero-adsb/dataero-adsb-feeder). Its installer does everything a receiver needs: it installs the decoder (readsb) if the Pi does not have one, registers your receiver with Dataero, generates its unique receiver ID, sets up the services that forward the messages and keep the connection alive, and checks at the end that Dataero can hear you. You can read every line of it on GitHub before you run it. Three commands, typed on the Pi, one after the other:

```
git clone https://github.com/dataero-adsb/dataero-adsb-feeder
cd dataero-adsb-feeder
sudo bash installer.sh
```

The installer asks a few questions. Here is what each one means and what to answer.

The installer asks	What it means	What to answer
Do you want to automatically install readsb? (yes/no)	No decoder was found on this Pi; readsb is the program that turns the dongle's samples into messages.	Type yes, in full. Any other answer stops the installer. The download and build take a few minutes.
Station name (optional)	The name shown with your receiver on the map and the leaderboard.	A short name: your town and a word, a school's name. It can be changed later by running the installer again.

The installer asks	What it means	What to answer
Enable MLAT for this receiver? (y/N)	Multilateration (chapter 10): your receiver's timing helps locate aircraft that send no position. It needs an exact antenna position.	n for a first install, or y if you can give the antenna's position to within a few metres.
Latitude, Longitude, Antenna altitude (metres)	Where the antenna is. Mandatory with MLAT, optional otherwise; used for your coverage map and range statistics.	Decimal degrees with a dot (50.8467, 4.3525), altitude above sea level in metres: the ground elevation at your address plus the antenna's height above the ground.

To find the position, open any online map, right-click the exact spot of the antenna and copy the coordinates; most maps show six decimals, which is a precision of a tenth of a metre. The ground elevation is on a topographic map or in the 'elevation' field of a GPS app; add the height of the antenna above the ground (a second-floor window is about 6 m, a roof ridge of a two-storey house about 9 m). A phone's own altitude is often ten or twenty metres wrong, so use the map for the ground and a tape measure for the rest. Do not enable MLAT with a guessed position: a receiver whose timing says 'I am here' from somewhere else degrades every solution it takes part in.

The installer then works for a few minutes and finishes with a banner that begins **Installation complete**. It prints your **receiver ID**, a long code of letters and digits (a UUID), the Dataero hub your station is connected to, and the result of a first contact with Dataero: when the registrar accepts the receiver's heartbeat, your station is live. The receiver ID is the key to your station: it travels with every message and is what you paste into your account to claim it. Keep it to yourself, because the first account that links it owns the receiver. If you ever lose it, it is stored on the Pi:

```
sudo grep RECEIVER_UUID /usr/local/dataero-adsb-feeder/.env
```

What the installer leaves running

Three small services that start by themselves at every boot: *readsb*, the decoder, which also serves a local map; *dataero-readsb*, a second, network-only decoder instance that forwards a reduced stream of your messages to Dataero's hub; and *dataero-feeder*, the heartbeat that tells Dataero your station is alive. With MLAT on, a fourth, *dataero-mlat*. Running the installer again later keeps your receiver ID, name and position and only updates what changed.

STEP 4

Going live, and claiming your receiver

Within a minute of the banner, the first aircraft should appear. Look in three places. On the Pi itself, the command below prints a live table of every aircraft being decoded, with its address, call sign, altitude, speed and the signal strength in dBFS; press **q** to leave it. The *readsb* installer also set up a local map on the Pi, <http://adsb.local/tar1090> (or the IP address instead of the name), which shows your own station's view of the sky with range rings. And on radar.dataero.eu, the public map, the aircraft your station hears are drawn with everyone else's within seconds.

```
viewadsb
```

Your receiver is already feeding, anonymously. **Claiming** it attaches it to your account and unlocks the feeder's side of the site: open radar.dataero.eu, create a free account or sign in, open **Receivers** in the account menu, choose **Link a receiver** and paste the receiver ID the installer printed. Use the Dataero receiver ID, not a UUID from another feeder program if you run one. From that moment your station has a name on the map, a **dashboard** (aircraft seen, messages per second, the busiest hours, the rarest aircraft), a **coverage map** drawn from the signal strength of every message in two and three dimensions, **365 days of flight history** instead of a visitor's none, and a rank on the **leaderboard**. Everything gathered before you claimed it is already there.

What to expect on the first evening: at a window with the box antenna, 20 to 60 aircraft at once in a busy part of Europe, out to 100 or 150 km, with a hole on the side of the building; with a tuned antenna in a loft, 100 to 200 aircraft out to 250 km; on a roof, 200 to 400 aircraft and a range that touches 400 km in the directions where nothing is in the way. The message counter in the dashboard is a better measure than the range: a good station decodes 500 to 1 500 messages a second at a European daytime peak and a few dozen in the small hours. The curve over a day is lesson 4 of the course made visible: aircraft do not fly at night because people do not.

Open the live map <https://radar.dataero.eu/map>

Create an account and claim <https://radar.dataero.eu/contribute>

STEP 5

Tuning, and understanding what you see

Two settings decide most of a station's performance, and both can be changed in a second on the Pi. The first is the **gain**, how much the dongle amplifies before it digitises. readsb starts with automatic gain, which is a fair compromise. If aircraft overhead vanish while distant ones remain, the receiver is overloaded by the strong nearby signals and the gain is too high; if the far aircraft never appear, it is too low. The command below sets a fixed gain in dB; try values between 30 and 49 (49.6 is the dongle's maximum) and watch the aircraft count and the maximum range on the local map for ten minutes each. With a dongle that has its own amplifier, 40 to 45 is usually best at a window and lower on a roof. Typing a value of -10 returns to automatic.

```
sudo readsb-gain 43
```

The second is the **location of the local map**, which readsb uses to draw its range rings and to compute each aircraft's distance; it is separate from the position you gave the Dataero installer (which is the one Dataero uses) and is set with the second command. Then open the local map and watch the outline of the furthest aircraft grow over a day: that outline is your radio horizon, and its shape is the shape of your obstacles.

```
sudo readsb-set-location 50.8467 4.3525
```

The signal column in *viewadsb* is the inverse square law of chapter 6 in numbers. An aircraft passing overhead at 10 km registers around -5 dBFS; the same aircraft at 300 km, thirty times further, arrives nine hundred times weaker, about -30 dBFS, and the decoder still reads it because the message only needs to be stronger than the noise. Watch an aircraft's signal rise and fall as it crosses your sky and you have measured the law without a formula. The column of dropped messages, when it is high, says the opposite: strong signals from two aircraft arriving at the same instant garble each other, which is the limit chapter 11 describes and the reason a second receiver thirty kilometres away sees messages that you lose.

If you enabled **MLAT**, the aircraft positioned by multilateration appear on the Dataero map and flow back into your local map with a distinct marker. These are the aircraft that send no position of their own: older business jets, military transports, some helicopters. Your contribution to each one is a timestamp accurate to a few nanoseconds, from which the hub solves the hyperbolae of chapter 10 with three other stations. A wrong antenna position of only 30 m shifts your timing by a tenth of a microsecond, enough to put a solved aircraft a kilometre off, which is why the installer insists.

WHEN IT DOES NOT WORK

Troubleshooting

Almost every problem is one of those in the table below, and each has a one-line check. Commands run on the Pi, over the same terminal connection as the installation.

Symptom	Most likely cause	What to do
ssh cannot find adsb.local	The router does not resolve local names, or the Pi is not on the network.	Use the IP address from the router's device list. If the Pi is not listed, check the Wi-Fi details you typed in Imager, or use an Ethernet cable.
readsb does not start; 'no device found'	The dongle is not seen: loose USB cable, a power supply too weak, or a dongle that is not an RTL-SDR.	Reseat the dongle and extension; run <code>lsusb</code> and look for 'RTL2838' or 'Realtek'; swap the power supply for the official one.
readsb runs but shows no aircraft	The antenna is not connected, the cable has an open connector, or the antenna is in a radio shadow.	Check every connector is tight; move the antenna to a window with open sky; a plain dongle without a filter in a city may need one.
Aircraft nearby disappear, far ones stay	Overload: gain too high, or a strong transmitter nearby.	Lower the gain (<code>readsb-gain 35</code>), add a 1090 MHz filter.
Aircraft on the local map but not on radar.dataero.eu	The feeder services are not running or the Pi has no internet.	Run the status command below; a service shown as 'failed' tells you why in its log. Run the installer again: it repairs the configuration and keeps your ID.
The Pi reboots or the dongle disappears at random	Power. The dongle draws a third of an ampere; a weak supply drops the voltage.	Use the official power supply; remove other USB devices; check the red power light stays steady.
'Receiver not found' when claiming	A different UUID was pasted (from another feeder program, or a typo).	Paste the value printed by the <code>grep</code> command in step 3; it is the Dataero receiver ID.

```
sudo systemctl status readsb dataero-readsb dataero-feeder
sudo journalctl -u dataero-feeder -n 50
```

If none of this helps, open an issue on the feeder's GitHub page with the output of those two commands: the people who wrote the installer read them. Dataero also runs a community channel linked from the Become a feeder page, where other feeders answer most questions within hours.

AFTERWARDS

Keeping it running

A receiver needs almost nothing. The services start by themselves after a power cut; the Pi has no moving parts; the dongle lasts years. Once a season, connect to the Pi and run the update line from step 2 again, then run the installer again from the folder it was downloaded into (*cd dataero-adsb-feeder* then *sudo bash installer.sh*): it fetches the current feeder, keeps your receiver ID, name and position, and asks the same questions with your previous answers as the defaults. If you move the antenna, run the installer again and give the new position; if you move house, do the same and nothing else changes. If you ever want to stop, the same folder contains an uninstaller.

Two habits keep a station healthy. Look at the dashboard once a week: a message rate that halves overnight means a cable has worked loose or a tree has grown. And write down, on the first day, three numbers from the local map: the maximum range in each direction, the number of aircraft at the evening peak and the message rate. They are your baseline, and every change you make afterwards, a better antenna, a higher mast, a filter, is measured against them. That measurement is the science the course was leading to.

FOR A CLASS OR A FAMILY

Turning the receiver into a project

The receiver is a permanent instrument, and the best use of it is a series of measurements that each test one idea from the course. For a family, one of these fills a Sunday; for a class, each one is a lesson, and the data the station gathers between lessons is the homework. The dashboard's hourly chart, the coverage map and the local map's range outline are the instruments; the lessons give the predictions; the measurements decide.

1. **Height against range** (chapter 7). Note the maximum range with the antenna on a table, at a first-floor window, at a top-floor window, in the loft. Plot range against height and compare with the radio horizon formula; the difference is the obstacles.
2. **The inverse square law** (chapter 6). Follow one aircraft across the sky in *viewadsb* and record signal against distance every minute; plot signal in dB against the logarithm of distance; the slope should be close to -20 dB per decade.
3. **The daily rhythm** (lesson 4). Read the message rate every hour for a week from the dashboard. Explain the shape: the morning wave, the evening peak, the night curfew at the nearest airport, the difference between a weekday and a Sunday.
4. **Where the sky ends**. Trace the range outline on the local map after a week and lay it over a map of the area. Each dent is a building, a hill or a tree; name them.

5. The map is only as good as its listeners (chapter 11). Compare your station's coverage map with the public map on an evening: the aircraft you do not hear are heard by someone else. Find the nearest other station on the leaderboard and work out, with the formula, where your circles overlap.

In a school, the receiver also answers the question every class asks, 'what is that plane?', with a screen in the corridor showing the local map. The dashboard's 'rarest aircraft' list is a monthly surprise: a government transport, a vintage airliner, an air ambulance, each one a story about why it is there. And a school station that feeds Dataero appears on the public map with the school's name as its source, which is a better kind of fame than most.

WELCOME

What feeders get, and what the map gets from you

A visitor to radar.dataero.eu sees a live map, a capped number of aircraft and a few fields per aircraft, for a viewing session of ten minutes at a time. A free account removes the session limit, shows every field and keeps seven days of flight history. A feeder, which is what you are now, gets the whole site: 365 days of history for every flight, the dashboard, the coverage map in two and three dimensions, the leaderboard, and a station credited on the map. None of it costs anything, now or later, and the receiver keeps feeding any other network you choose to add.

What the map gets from you is the thing no money buys: a listener where there was none. Every receiver on the network hears a circle of sky, and a continent is covered only where those circles overlap. A new station in a gap, a village, a coast, a valley, a country with few feeders, makes aircraft appear that nobody was hearing and makes multilateration possible for the ones that send no position. Look at the coverage of your region on the map tonight, and then look at it a week after your station is live: the difference is yours.

Become a feeder <https://radar.dataero.eu/contribute>

Open the live map <https://radar.dataero.eu/map>

Back to the course home <https://radar.dataero.eu/learning>