LK-29

Geiger Counter & Weather Station

User Documentation

Version 1.4.0 • July 2026

1. Introduction

LK-29 is an intelligent weather station with an ionizing-radiation monitor, air-quality indicator, web interface, and support for storing measured values in InfluxDB. It runs on an ESP32-S3 microcontroller with firmware written in MicroPython. The system is designed for continuous low-power operation, remote configuration through the web interface, and OTA firmware updates.

The device combines ionizing-radiation detection (GM tube + high-voltage regulator) with environmental monitoring (BME688 - temperature, humidity, pressure, air quality) and provides several visualization methods: a 2.9" e-Paper display, responsive web dashboard, real-time console, weather-service integration (OpenWeatherMap, Open-Meteo), and InfluxDB logging.

1.1 New and important in version 1.3.5

The /config, /wifi, /ota and /console admin pages share a unified blue color palette while keeping their original layout and dark/light theme switching.
BME688 pressure is converted to sea-level pressure immediately after each reading, using the station altitude.
Added the STATION_ELEVATION_M configuration value; for coordinates 48.66690000, 17.53474600 the default value is 268 m above sea level.
Station altitude is automatically refreshed from OpenTopoData SRTM90m when the configuration is saved if the location coordinates changed or no altitude is configured yet.
config.PRESSURE_RAW keeps the absolute sensor pressure, while config.PRESSURE used by the tile, chart, history, and InfluxDB contains pressure reduced to sea level.

- Fully functional IAQ index from the BME688 gas sensor with controlled baseline normalization and long-term clean-air trend tracking
- Mild compensation for temperature and humidity influence on IAQ measurement, adapted for outdoor conditions
- More accurate gas resistance measurement (Bosch compensation) and a smoother IAQ curve without step jumps (“teeth”) - see the Addendum at the end of this document

Important IAQ limitation: the BME688 gas sensor is designed mainly for indoor air-quality estimation. It does not measure particulate matter such as PM1, PM2.5 or PM10, nor dust, pollen or aerosol concentration. In outdoor use, humidity, temperature, airflow and sensor ageing can influence the gas resistance strongly, so the displayed IAQ value should be treated as an indicative trend only.

- Colored dashboard tile backgrounds based on measured values (20% intensity)
- Drag-and-drop chart configuration: drag tiles directly to the left or right chart axis
- Per-browser chart configuration storage (each client remembers its own settings)
- JavaScript-driven font sizing - uniform font size across all metric tiles
- Acoustic signaling of GM pulses (passive buzzer on GPIO10)
- OTA firmware upgrade through the /ota web interface with atomic ZIP extraction
- Real-time upgrade streaming progress (chunked transfer encoding)
- Web console at /console - live ESP32 output with YYYY-MM-DD HH:MM:SS timestamps
- Weather forecast with Moon phases (smooth SVG rendering)

Weather forecast icons and temperature labels automatically resize after portrait/landscape changes and browser window resizing.
Tap or click a metric tile to show the last-24-hour minimum and maximum value in the tile for 7 seconds, including the measurement unit.
Dashboard chart drag-and-drop now uses pointer coordinates with long-press support, improving reliability on Android tablets and browsers with custom touch handling.

- Two weather sources: OpenWeatherMap and Open-Meteo, switchable through the web UI
- InfluxDB records include id (MAC address) and ip (current IP address) tags

- E-paper display rotated 180 degrees + micro symbol glyph at position 128, SK/EN localization

1.2 Hardware specifications

Component	Specification
MCU	ESP32-S3 Zero (240 MHz dual-core, 8 MB PSRAM, 4 MB Flash)
Display	Waveshare 2.9" e-Paper B V3 (128x296 px, black/red/white)
GM tube	XERAM 3G8B (pulses to GPIO1) - wide-range industrial tube, 445 V threshold
BME688	I2C 0x77 (temperature, humidity, pressure, IAQ gas sensor)
High-voltage regulator	PWM-controlled boost converter, target Uout = 400-450 V
Buzzer	Passive, 3 kHz PWM on GPIO10 (tick on each GM pulse)
Power supply	USB-C 5V or Li-Ion battery + boost (3.7V -> 5V)
Consumption	~150 mA average, ~250 mA during Wi-Fi transfer
Wi-Fi	802.11 b/g/n (2.4 GHz, the only supported frequency)
Operating temperature	0 °C to +50 °C
Dimensions	~80 x 50 x 25 mm

1.3 ESP32-S3 Zero pinout

GPIO	Function
1	GM tube input
2	Battery voltage divider (ADC)
3	SPI MOSI (e-paper)
4	SPI CLK (e-paper)
5	Display RST
6	HV ADC (high-voltage feedback)
7	PWM output for HV regulator
8	I²C SCL (BME688)
9	I²C SDA (BME688)
10	Buzzer PWM output
11	Display BUSY (input)
12	Display DC (Data/Command)
13	Display CS (Chip Select)
16	SPI MISO (unused)

2. Dashboard

The dashboard is the main user screen available at http://<ip>/ or http://<ip>/dashboard. It is optimized for both mobile devices and desktop computers (responsive design). It does not require login and refreshes approximately every 15 seconds using /api/data polling.

2.1 Top panel

The top part of the screen shows time information and the wind compass:

Item	Description
Time (large)	Current time in 24-hour format (hh:mm), synchronized via NTP
Day	Weekday name (for example Monday) according to LANG
Date	Full date in local format
Wind compass	Arrow indicating wind direction (north up), with speed in m/s below
Theme	Sun/moon button - switches dark/light mode (localStorage)

2.2 Metric tiles (6)

Six tiles show current parameter values. Font size is calculated dynamically by the JavaScript fitMetricTiles() function.

Tile	Unit	Data source	**Draggable**
Temperature	°C	BME688 / SHT30 (I²C)	Yes
Humidity	% RH	BME688 / SHT30 (I²C)	Yes
Pressure	hPa	BME688 (I2C), reduced to sea level	Yes
Wind	m/s	OpenWeatherMap / Open-Meteo	No
Radiation	µSv/h (auto)	GM tube + CPM calculation	Yes
Air quality (IAQ)	0–500	BME688 gas sensor	Yes

2.2.1 Colored tile backgrounds

Each tile background changes dynamically according to the current value (20% intensity = alpha 0.2).

2.2.2 Temperature (tempColor)

Range	Background color
below -10 °C	dark blue (frost)
-10 to 0 °C	light blue (cold)
0 to 15 °C	dark green (spring/autumn)
15 to 24 °C	light green (pleasant)
24 to 33 °C	orange (warm)
above 33 °C	red (heat)

2.2.3 Humidity (humColor)

Range	Background color
below 50%	green (dry air)
>= 50%	blue (humid air)

2.2.4 Wind (windColor)

m/s range	Background color
0 - 3	pale green (calm)
3 - 6	green (light breeze)
6 - 15	pale blue (fresh wind)
15 - 30	orange (strong wind)
above 30	red (gale)

2.2.5 Pressure (presColor)

Starting with version 1.3.5, the pressure tile and chart show pressure reduced to sea level. The absolute local pressure from the BME688 sensor remains available internally as PRESSURE_RAW, while PRESSURE used by the web UI, history, and InfluxDB is calculated from STATION_ELEVATION_M.
Tap or click a metric tile to temporarily replace the current value with two lines: min value unit and max value unit. The overlay uses the same 24-hour interval as the charts and disappears automatically after 7 seconds.
Supported tiles are Temperature, Humidity, Pressure, Radiation and Air quality. Wind has no history buffer and therefore does not show a min/max overlay.

hPa range	Background color
below 1000	blue (low pressure - stormy weather)
1000 – 1020	green (normal pressure)
above 1020	orange (high pressure - anticyclone)

2.2.6 Radiation (radColor)

Range	Background color
below 1 µSv/h	green (safe)
1 – 10 µSv/h	orange (elevated)
above 10 µSv/h	red (high)

2.2.7 Air quality (iaqColor)

Smooth gradient through three colors - green -> blue -> red - interpolated by IAQ value:

IAQ is an orientation value from the BME688 gas sensor. It is not a dust/particulate measurement and must not be interpreted as PM2.5, PM10 or a calibrated public air-quality index.

IAQ	Background color
0 (Excellent)	deep green (clean air)
150 (Moderately polluted)	blue (average air)
300+ (Hazardous)	red (strong pollution)

2.3 Configurable charts

The dashboard shows two charts, each with two metrics (left and right Y-axis). Version 1.3.5 keeps the flexible drag-and-drop configuration directly from metric tiles:

2.3.1 How to change chart metrics

1. Grab any metric tile (Temperature, Humidity, Pressure, Radiation, Air quality) - the cursor changes to "grab"
2. Drag with the mouse, or press and hold a metric tile on a touchscreen and then drag it. A small movement threshold prevents normal taps from being treated as chart drag operations.
3. Two drop zones appear above the charts: the left half for the left Y-axis and the right half for the right Y-axis

On touch devices, long-press handling and disabled browser context menus make drag-and-drop more reliable in Chrome, Adblock Browser and older tablet browsers.

4. Drop the tile onto the selected half of the selected chart
5. The chart is redrawn immediately with the new metric and the legend is updated

2.3.2 Auto-swap on duplicate

If you drag a metric that is already shown in another chart, the system automatically performs a swap: the original metric in the target slot moves to the freed slot, so each metric is displayed only once.

Example: Chart 1 = Temperature + Humidity, Chart 2 = Pressure + Radiation. If you drag Pressure to the left axis of Chart 1, the result is: Chart 1 = Pressure + Humidity, Chart 2 = Temperature + Radiation.

2.3.3 Per-browser configuration

Chart configuration is stored in browser localStorage (key lk29-chart-cfg):

- Each device / browser has its own configuration
- A phone can show Temperature + IAQ while a PC shows Pressure + Radiation - independently
- The server (ESP32) does not know the configuration - no disk I/O and no OTA risk
- Reset: in Dev Tools (F12), type localStorage.removeItem('lk29-chart-cfg') and refresh
- After refresh, default values are used: Chart 1 = Temperature + Humidity, Chart 2 = Pressure + Radiation

2.3.4 Chart properties

- Shows the last 24 hours (1 point per minute, max. 1440 points)
- Smooth rendering (Bezier spline, tension 0.2)
- Dynamic axis scaling (auto-fit min/max with 5% padding)

- Dual Y-axis: left axis for the first metric, right axis for the second
- Y-axis labels use colors matching the chart lines
- History is stored in ESP32 RAM (history.py - circular buffer with 1440 samples)
- After ESP32 restart, history is lost (it is not persistent)

2.4 Weather forecast with Moon phase

The lower part of the dashboard shows a 5-day weather forecast from OpenWeatherMap or Open-Meteo:

- Day of week (abbreviated according to LANG)
- Month and day (for example 17 May)
- SVG Moon phase icon (smooth rendering, not emoji)
- Weather icon (sun, partly cloudy, cloud, rain, thunderstorm, snow, fog)

Forecast weather icons are scaled from the actual forecast-tile geometry and are recalculated after resize and orientation changes. This prevents oversized or stale icon sizes when switching between portrait and landscape.

- Maximum temperature (bright) and minimum temperature (muted)

Moon phase is calculated from the date (synodic period 29.53 days) and shown as smooth SVG, not as a discrete emoji.

2.5 Display modes

Mode	Characteristics
Portrait (mobile)	Vertical stack: top-left -> metrics 2x3 -> charts -> forecast
Landscape (tablet)	50/50 split: clock+compass on the left, 3x2 tiles on the right
Desktop (wide screen)	Full layout with large tiles and wide forecast
Phone landscape (21:9)	Clock in narrow font, optimized for extremely wide displays

3. Configuration panel

Available at http://<ip>/config. Requires login (HTTP Basic Auth):

- Username: admin (fixed)
- Password: 1423 (default - we recommend changing it immediately via Change PWD)

The panel shows 6 sections. Chart configuration is NOT in the panel; it is handled by drag-and-drop directly on the dashboard (see section 2.3).

3.1 Live Measurements

Current measured values (read-only, refreshed every 5 s):

- Battery Voltage, HV Voltage, Duty Cycle
- GM Pulses, CPM, Dose Rate
- Temperature, Humidity, Pressure (sea level)
- IAQ + IAQ text

3.2 Wi-Fi Status

- SSID + signal strength (dBm)
- IP, MAC address, hostname
- NTP synchronization, DNS servers

3.3 I2C Bus

- I2C device addresses (hex)
- Detected device type (BME688, SHT30, etc.)
- Re-scan button

3.4 Read-Only Configuration

- Firmware version, uptime
- Time and date (timezone + DST)
- Location (GeoIP)
- RAM/Flash usage, CPU frequency

3.5 OpenMeteo / OpenWeatherMap

Current weather from the active source, switchable by checkboxes in Editable Configuration.

3.6 Editable Configuration

Configuration parameters are divided into groups with colored dividers:

Group	Color	Parameters
Firmware	Orange	OTA upgrade link
OpenMeteo / OpenWeatherMap	Blue	OWM/OM checkbox, API key, GPS
BAT divider	Orange	VBAT_DIVIDER_RATIO
HV Regulator	Gray	TARGET_VOLTAGE, DEADBAND, PID coeff., MAX_DUTY
Buzzer	Purple	BUZZER_ENABLED, ACOUSTIC_TICK
Radiation	Brown	CPM_PER_USV, CPM_ALARM_THRESHOLD, CPM_PERIOD_MS
I2C / Sensors	Gray	I2C_INTERVAL, REFRESH_DISP
Time	Blue	TIMEZONE_DEV, DST_DEV, SYNC_INTERVAL, LANG
InfluxDB	Purple	INFLUX_ENABLED, URL, BUCKET, ORG, TOKEN, INTERVAL
Debug	Gray	DEBUG, DEBUG_LEVEL bitmask

3.6.1 Weather source switching

The OpenMeteo / OpenWeatherMap section contains two mutually exclusive checkboxes:

- Checking one automatically unchecks the other
- The change is saved to config.py immediately (only these 2 values, not the full config)
- The Restart button turns red - indicating that a restart is needed
- Both checkboxes can be unchecked - weather fetching will be disabled

3.6.2 Sensitive field masking

API Key OWM and InfluxDB Token are masked in the configuration panel:

- Placeholder "*** (token is set)" is shown instead of the actual value
- When saving, an empty field preserves the existing value (it is not overwritten)
- To change it: delete the placeholder, enter a new value, click the check mark or Save

3.7 Buttons

Button	Function
Restart	Restart ESP32 (confirmation dialog)
Change PWD	Change password (2x confirmation)
Save config.	Atomic write of all changed editable variables to config.py
Download config.	Download current config.py from the device (backup)
⬆ Upgrade	Navigates to the /ota firmware upgrade page

3.8 Theme

The sun/moon button in the top-right corner switches dark/light mode (stored in localStorage).

4. Program behavior

4.1 Startup (boot)

6. Boot ESP32-S3 (~1 s) - MicroPython runtime initialization
7. main.py - bootstrap, loading config.py
8. Wi-Fi connection (~2-5 s)
9. On failure: AP fallback (SSID GeigerCounter), captive portal
10. NTP sync (~1 s)
11. GeoIP lookup (~1 s)
12. HTTP server start (~1 s)
13. E-paper init (~3 s)
14. I2C scan (~1 s)
15. Main loop - measurement + rendering

4.2 High-voltage regulation

The HV regulator uses a PI controller to maintain the target voltage:

- ADC reads the voltage feedback (1:1000 divider)
- Target value TARGET_VOLTAGE (typ. 450 V)
- Deadband prevents duty-cycle jitter
- PWM duty is adjusted proportionally to the error
- PWM frequency: 100 kHz
- Max duty: MAX_DUTY (typ. 70%) - safety limit

4.3 CPM and dose-rate calculation

- IRQ handler on GPIO1 (rising edge) increments the counter on each pulse
- micropython.schedule() is called from IRQ context for the buzzer tick
- CPM = rolling count over the last 60 seconds
- Dose rate (uSv/h) = CPM / CPM_PER_USV
- When CPM_ALARM_THRESHOLD is exceeded, the acoustic alarm starts
- During rain the dose rate can temporarily rise - raindrops scavenge atmospheric radon decay products (mainly Pb-214 and Bi-214) out of a large volume of air and deposit them near the ground (so-called radon washout), a well-documented and normal phenomenon, not a sensor fault. The increase typically fades gradually over the following hours.

4.4 I2C sensors

BME688 / BME280 / SHT30 are read every I2C_INTERVAL seconds:

- Auto-detection by I2C scan at startup
- Forced-mode measurement (one-shot) for power saving
- BME688 gas measurement: custom triggering (run_gas bit 5)

4.4.1 IAQ algorithm

The air quality index is calculated in 4 steps:

16. BME688 gas resistance (ohms) is read from registers 0x2A-0x2B
17. Baseline handling: IAQ_GAS_BASELINE is adjusted only by quiet-window normalization after stable trusted gas samples and acceptable humidity conditions
18. IAQ calculation: ratio = log10(baseline / gas), IAQ = 10..500 according to the ratio
19. Mild compensation using clipped T and H values

4.4.2 Why not the original Bosch coefficients

The original Bosch coefficients (2.50 for humidity, 0.52 for temperature) are designed for the closed BSEC algorithm and calibrated for INDOOR environments with 30-60% RH. For outdoor use with 80-95% humidity, they cause a serious problem:

- At H = 87%, the correction is 2.50 x (87-40) = 117.5 IAQ
- With clean air (IAQ_raw about 14), the result is 14 - 117.5 = -103 -> clamped to 5
- The sensor gets stuck at the false value IAQ = 5 (false-clean)

LK-29 therefore uses a modified formula with three safeguards:

- Range clipping: H is clipped to 30-85%, T to 5-35 deg C
- Coefficients reduced 5x: 2.50 -> 0.50 and 0.52 -> 0.10
- Higher threshold: compensation is applied only when ratio > 0.10

Real case: when outdoor humidity rose from 81% to 87% at night, the original formula caused IAQ to drop from 11 -> 14 -> 5 (false-clean). After the fix, the new formula gives IAQ = 15 under the same conditions (a realistic value).

4.5 Weather (OpenWeatherMap / Open-Meteo)

Source	Protocol	API key	Forecast
OpenWeatherMap	HTTP	Required	5 days / 3h
Open-Meteo	HTTPS (TLS)	Not required	16 days / 1h

Fetch interval: OWM_INTERVAL (default 1800s = 30 min).

4.6 InfluxDB

- Measurement: lk29
- Tags: device_id (MAC address), ip (current IP address)
- Fields: temperature, humidity, pressure, iaq, cpm, usv_per_hour, hv_voltage, battery_voltage
- Support for both InfluxDB 1.x and 2.x

4.7 ePaper display

Waveshare 2.9" B V3 (128x296 px) refreshes every REFRESH_DISP seconds:

- 3-stage initialization
- 180-degree rotation (the panel is physically upside down)
- Built-in 5x7 ASCII font + degree sign + micro sign
- Localization from lang_sk.json / lang_en.json
- 3 colors: black, red, white
- Refresh takes about 3 s - the display flashes during a full update

4.7.1 Screenshot

At http://<ip>/screenshot, a BMP image of the current ePaper display content is generated (128x296 px, 3-color, ~111 KB). Does not require login.

4.8 OTA Firmware Upgrade

The /ota web page allows uploading a firmware ZIP archive without physical access:

4.8.1 Phase 1: Upload

20. The client selects a .zip file
21. The client calculates CRC32

22. POST /api/ota/upload with X-CRC32 header
23. The server stores the file and verifies CRC32

4.8.2 Phase 2: Extract & Run

24. GET /api/ota/run - the server starts extracting the ZIP archive
25. The response is a chunked text/plain stream
26. After completion: RESTART -> the device restarts

4.9 Web Console

Endpoint /console shows live ESP32 output through regular polling (/api/console/recent, every 1 s):

- Timestamp YYYY-MM-DD HH:MM:SS for every line
- Color by type: info white, warn yellow, err red, dbg gray
- Filter buttons for individual modules
- Max 1000 lines in the buffer

4.10 Web interface

Address	Auth	Description
/ or /dashboard	No	Main dashboard
/config	Yes	Configuration panel
/ota	Yes	OTA firmware upgrade
/console	Yes	Live console log stream
/wifi	No	Wi-Fi setup (captive portal)
/screenshot	No	BMP display screenshot
/api/data	No	JSON with current measurements and config
/api/history	No	JSON with 24h history
/api/lang/sk	No	JSON with SK translations
/api/lang/en	No	JSON with EN translations
/api/set?key=val	Yes	Update one variable in RAM
/api/save	Yes	Atomic config.py write
/api/restart	Yes	Restart device
/api/password	Yes	Change password
/api/ota/upload	Yes	POST firmware ZIP
/api/ota/run	Yes	Extraction streaming
/api/console	Yes	SSE stream of log lines

4.11 HTTP authentication

The system uses HTTP Basic Auth (RFC 7617):

- By default without password (AUTH_PASSWORD = "") - everything is accessible
- After setting a password via "Change PWD" - protected endpoints require login
- Username: admin (fixed), password: user-defined

- The browser remembers login during the session
- Forgotten password: run reset_pwd.py through the serial port (Thonny)

5. Program concept

5.1 Modular architecture

Module	Function
main.py	Main control loop
config.py	Global variables, persistence (atomic write)
reg.py	HV regulation, ADC, GM pulse IRQ, CPM calculation
buzzer.py	Acoustic signaling
bme680.py	robert-hh library for BME688 (forked, modified)
i2c.py	I2C scan + driver + IAQ algorithm
display.py	E-paper rendering
epaper2in9bv3.py	Low-level Waveshare driver
wifi_manager.py	Wi-Fi scan, AP fallback, captive portal
http_server.py	HTTP server router + _hw_map builder
http_api.py	REST API endpoints (/api/*)
http_cfg.py	Editable variables with r_ prefix
ota.py	OTA upgrade engine
history.py	24h circular buffer
influx.py	InfluxDB client
openweather.py / openmeteo.py	Weather API clients
geoip.py	GeoIP lookup
console.py	SSE real-time log stream

5.2 Configuration management

- config.py - main configuration file (atomic write using .tmp + rename)
- http_cfg.py - variables with the r_ prefix are editable through the web UI
- http_server.json - defines the structure of the web interface
- wifi_config.json - SSID, password, hostname, static IP

Chart configuration is NOT in config.py - it is stored in browser localStorage (per-browser).

5.3 Debugging

Bit	Value	Module
0	0x01	main.py
1	0x02	reg.py
2	0x04	wifi_manager.py
3	0x08	http_*.py
4	0x10	i2c.py + bme680.py
5	0x20	display.py
6	0x40	geoip.py
7	0x80	openweather/openmeteo.py
8	0x100	influx.py

6. Configuration parameters (meaning)

6.1 HV regulator

Parameter	Meaning	Typical range
TARGET_VOLTAGE	Target voltage (V)	100 – 600
DEADBAND	Deadband (V)	0.005 – 0.02
KP	Proportional gain	0.05 – 0.5
KI	Integral gain	0.01 – 0.1
KD	Derivative gain	0 – 50
MAX_DUTY	Maximum PWM duty (%)	50 – 80
REGULATOR_PERIOD_MS	PI regulator period (ms)	50 – 500

6.2 Battery

Parameter	Meaning
VBAT_DIVIDER_RATIO	Voltage-divider ratio (R1+R2)/R2

6.3 Radiation

Parameter	Meaning
CPM_PER_USV	Pulse count per 1 uSv/h - depends on the tube model (currently XERAM 3G8B: 6.85, derived from the datasheet: "1 stroke/sec per mR/h")
CPM_ALARM_THRESHOLD	Alarm threshold (uSv/h)
CPM_PERIOD_MS	CPM calculation interval (ms)

6.4 Buzzer

Parameter	Meaning
BUZZER_ENABLED	Main acoustic switch
ACOUSTIC_TICK	Tick on each GM pulse

6.5 I2C / Sensors

Parameter	Meaning
I2C_INTERVAL	Sensor reading interval (s)
REFRESH_DISP	Display refresh interval (s)

6.6 Time

Parameter	Meaning
TIMEZONE_DEV	UTC offset (-12 to +12)
DST_DEV	Daylight saving time (True/False)
SYNC_INTERVAL_DEV	NTP synchronization interval (h)
LANG	UI language (sk/en)

6.7 Weather source

Parameter	Meaning
OWM_ENABLED	Use OpenWeatherMap
OM_ENABLED	Use Open-Meteo (mutually exclusive with OWM)
OWM_API_KEY	OpenWeatherMap API key
OWM_LAT, OWM_LON	GPS coordinates for weather and altitude lookup
OWM_INTERVAL	Fetch interval (s)
STATION_ELEVATION_M	Station altitude in meters; used to reduce pressure to sea level and refreshed automatically from OpenTopoData SRTM90m when location changes/configuration is saved

6.8 IAQ (Air quality)

The BME688 IAQ value is useful for observing relative changes and trends, especially after baseline calibration. The sensor is intended primarily for indoor use and does not detect particulate matter, so outdoor IAQ readings should be considered approximate and informative only.
Why IAQ often worsens during rain: humidity plays a part, and the sensor only partially compensates for it, but that is not the only reason. Rain hitting dry soil, pavement, or plants releases real volatile compounds accumulated during dry weather (so-called petrichor) - a genuine chemical signal, not a measurement error. This is often compounded by the lower atmospheric pressure and weaker air mixing typical of rainy weather, so gases near the ground disperse less and build up closer to the sensor. In very high humidity or fog, a thin film of water can also form on the sensor's protective membrane, temporarily restricting gas diffusion to the sensing element. The IAQ drop after rain usually fades gradually.

Parameter	Meaning
IAQ_GAS_BASELINE	Auto-calibration reference (ohms), default 50000

6.9 InfluxDB

Parameter	Meaning
INFLUX_ENABLED	Enable InfluxDB
INFLUX_URL	Database URL
INFLUX_ORG, BUCKET, TOKEN	InfluxDB 2.x authentication
INFLUX_POINT	Measurement name
INFLUX_LOCATION	Location tag

INFLUX_INTERVAL	Write interval (s)
INFLUX_BUF_SIZE	Ring buffer size for pending records

6.10 Debug

Parameter	Meaning
DEBUG_LEVEL	Module bitmask (see 5.3)

6.11 HTTP authentication

Parameter	Meaning
AUTH_PASSWORD	Password for Basic Auth (empty string = no auth)

7. Troubleshooting

Wi-Fi setup shows 'Failed to fetch' or remains on 'Connecting'

The setup portal uses an asynchronous connection flow. After pressing Connect, the page sends /api/connect and then polls /api/connect_status. If the browser temporarily loses the AP connection while ESP32 switches Wi-Fi mode, keep the setup page open and wait for the status update.
If connection still fails, verify that the selected SSID is a 2.4 GHz network and re-enter the password.
A missing wifi_config.json is not fatal: the device starts the GeigerCounter AP and opens the setup portal.
Saved passwords are reused only for the same saved SSID. When selecting a different SSID, the portal requires a new password to avoid trying the old password on the wrong network.

Problem	Possible solution
Dashboard does not show values	Check Wi-Fi. Restart the device. /api/data must return JSON.
Drag & drop does not work on mobile	HTML5 drag-and-drop has weak touch support. Try from a PC.
Chart shows wrong metrics after refresh	Reset: localStorage.removeItem('lk29-chart-cfg') and refresh.
Charts are empty	Wait 1-2 min for data accumulation. History is lost after restart.
Wi-Fi does not connect	2.4 GHz only! Check SSID/password (case-sensitive).
Display remains blank	Check USB-C power supply (min 500 mA).
BME688 is not detected	I2C wiring (SCL=GPIO8, SDA=GPIO9). Re-scan in the config panel.
IAQ still shows 5	At first startup, allow 4-6h calibration. Reset baseline to 50000.
Buzzer does not beep	BUZZER_ENABLED=True. Check GPIO10 + GND.
OWM does not work	Check the API key. The free tier has limits.
InfluxDB does not write	Check URL, BUCKET, ORG, TOKEN.
OTA upgrade fails	ZIP must have valid CRC32. Min. 200 KB free.
Forgotten password	Run reset_pwd.py from Thonny IDE.

8. Deployment

8.1 Files to upload

Root directory (/):

```
config.py, main.py, reg.py, buzzer.py, bme680.py, wifi_manager.py, http_server.py,
http_api.py, http_cfg.py, i2c.py, geoip.py, openweather.py, openmeteo.py, history.py,
influx.py, ota.py, console.py, display.py, epaper2in9bv3.py
```

html/ directory:

```
dashboard.html, dashboard.css, main.html, wifi.html, ota.html, console.html,
http_server.json, http_server.css, manifest.json, favicon.svg, wind.svg, lang_sk.json,
lang_en.json, dashboard1.html, dashboard1.css, dashboard2.html, dashboard2.css,
dashboard3.html, dashboard3.css
```

8.2 First startup

27. Connect ESP32 via USB
28. Use Thonny IDE or ampy to upload all files
29. Restart ESP32 (Ctrl+D in Thonny)
30. The device starts the Wi-Fi AP "GeigerCounter"
31. Connect to this AP
32. The captive portal opens automatically, or open http://192.168.4.1
33. Select your home Wi-Fi and enter the password

Only 2.4 GHz Wi-Fi networks are supported by ESP32-S3. If a 5 GHz SSID is selected, connection will fail even when the password is correct.
The Wi-Fi portal sends the connection request immediately, then polls /api/connect_status until success or failure. This avoids browser fetch timeouts while the device is switching from AP mode to STA mode.

34. After saving, the device restarts and connects
35. The IP address is shown on the e-paper display
36. Open http://<ip>/config - set the password and API key

8.3 Important notes

- Before uploading new .py files, delete old .mpy files
- CSS files are cached - clear the cache after changes (Ctrl+Shift+R)
- OTA preserves persistent local settings and learned IAQ calibration values; differential upgrades use the saved OTA manifest after the first full upload
- Changing LANG requires a restart before it appears on the display
- When moving the device indoor/outdoor, reset IAQ_GAS_BASELINE to 50000
- The e-paper display has a limited lifetime (~1M refreshes) - about 2 years at a 60 s interval
- Chart configuration is stored in localStorage - when changing browser it resets to default

8.4 Installing as PWA

Android (Chrome)

37. Open http://<ip> in Chrome
38. Menu (three dots) -> "Install app"

iOS (Safari)

39. Open http://<ip> in Safari
40. Share -> "Add to Home Screen"

Desktop (Chrome/Edge)

41. Install icon (+) in the address bar
42. Click -> "Install"

9. Conclusion

LK-29 is a complete monitoring system suitable for home and professional use. Version 1.3.5 adds correct atmospheric-pressure handling and keeps the flexible user interface - drag-and-drop chart configuration directly from tiles and per-browser settings that allow different clients (phone, tablet, PC) to show different metrics as needed.

The modular architecture allows easy extension with additional sensors or features. The open-source character of the project (MicroPython + documented code) allows customization for specific needs.

Thanks to the OTA upgrade system, firmware can be updated without physical access - ideal for deployed devices.

9.1 Contact and support

Item	Value
Author	Ľubomír Karlik sr.
Website	www.dedeideas.eu
Software version	1.3.6
Release date	July 2026
Platform	ESP32-S3 + MicroPython 1.21+

v1.3.5 Addendum - IAQ Gas Normalization and 24 h Trend

Different BME688 modules can report gas resistance values that differ by orders of magnitude in the same clean air. The firmware therefore uses GAS_NORMALIZED so readings remain comparable after sensor replacement and between LK-29 and LK-30 devices.
The normalization formula is: GAS_NORMALIZED = GAS_RESISTANCE * GAS_NORM_TARGET / IAQ_CLEAN_BASELINE. The default GAS_NORM_TARGET is 250000 ohm. After 24 hours of stable clean air, IAQ_CLEAN_BASELINE is treated as the reference for the installed sensor.
A gas measurement is trusted only when BME688 reports gas_valid=True, heat_stab=True and the resistance is within 1000 to 50000000 ohm. Invalid or unstable readings must not update the IAQ baseline, clean-air vector, or trend state.
After IAQ_CLEAN_BASELINE changes, the firmware collects a new independent 24 h clean-air vector in iaq_clean_vector.json. The current 24 h normalized-gas window is then compared with that reference vector using both level change and curve shape.
A normalized-gas drop above 10% is treated as slow air-quality degradation. A smaller drop with a very similar curve shape may be marked as likely sensor drift. The status is available through IAQ_TREND_STATUS, IAQ_TREND_DELTA_PCT, IAQ_TREND_LEVEL_RATIO, and IAQ_TREND_SHAPE_ERROR_PCT.
Measurement history and diagnostics also include gas_normalized, gas_adc, gas_range, gas_valid, heat_stab, and gas_history_trusted. These fields help separate a real gas event from an unstable or incorrectly read BME688 measurement. Only trusted gas_normalized samples are allowed to update the clean-air vector and IAQ trend state.
BME688 is primarily intended for indoor environmental sensing. It does not measure particulate matter PM1/PM2.5/PM10, so IAQ must be treated as an indicative value, not as a certified air-quality measurement or a safety gas detector.

v1.3.5 Addendum - Bosch Gas Resistance Compensation and IAQ Curve Smoothing

Gas resistance conversion from the raw gas_adc/gas_range registers used a simplified estimate, var1 = 262144 >> gas_range, which ignored the per-chip range_switching_error calibration byte (register 0x04, already read into sensor._sw_err in bme688.py) and the Bosch-calibrated lookup tables for the 16 gas ranges. For certain gas_range values this simplified estimate deviated significantly from the chip's actual calibration, so every time the sensor's own autoranging switched range (a normal part of BME688 operation) the computed resistance jumped by a real, repeatable amount - visible as regular "teeth" on the IAQ/Gas history chart. This was not sensor noise but an algorithmic error.
Fix: gas resistance is now computed with the official Bosch formula using the LOOKUP_TABLE_1/LOOKUP_TABLE_2 tables and sensor._sw_err - the same formula already present (but never called, due to its blocking read) in the gas property in bme688.py.
Related second fix: the IAQ curve interpolation now genuinely runs over log10(baseline/gas ratio), exactly as already described in section 4.4.1 of this document - previously the calculation interpolated over the raw ratio itself, not its log10. The first curve segment (ratio 1.0 to IAQ_RATIO_100, default 2.0) therefore had a several-times steeper slope than later segments, amplifying ordinary sensor noise in that region into visible "teeth" and an uneven IAQ decline. IAQ values at the configured breakpoints (IAQ_RATIO_100..500) remain unchanged; only the shape of the curve between them changes.
Consequence for existing devices: the stored IAQ_GAS_BASELINE and IAQ_CLEAN_BASELINE values were calibrated under the old formula, so after this fix they temporarily no longer match the new (more accurate) resistance scale until they are re-settled by automatic normalization or a manual service baseline calibration.

Both fixes were originally identified and deployed on the parallel LK-30 airQ project (same BME688 fingerprint/IAQ foundation) and subsequently ported to LK-29.

v1.3.5 Addendum - OTA Manifest and Trusted Gas History

OTA differential upgrade now follows the LK-30 model. After a successful upgrade, the firmware stores /ota_manifest.json with the complete file list of the installed release. The /api/ota/manifest endpoint returns this saved manifest instead of walking the flash filesystem and calculating CRC32 for every file before each upload. If no manifest exists yet, the browser automatically uses a full upload; subsequent upgrades can upload only changed files.
The browser sends X-OTA-Mode: diff or full during /api/ota/upload, and the selected mode is kept in the OTA status for /api/ota/run. OTA_CLIENT_IDLE_TIMEOUT_MS and OTA_RUN_TIMEOUT_MS define the browser-side idle tolerance and server-side run timeout. Persistent local files, Wi-Fi settings, password, InfluxDB/OpenWeatherMap configuration and IAQ calibration values remain preserved across the upgrade.
IAQ trend handling was tightened. The runtime flag GAS_HISTORY_TRUSTED marks whether the current BME688 gas sample may feed gas_normalized, the clean-air vector and long-term trend detection. Clean-pulse samples, post-pulse settling samples, ADC-saturated values and readings with invalid gas_valid/heat_stab status remain visible as diagnostics, but they do not update the IAQ baseline, clean-air vector or trend state. /api/data exposes gas_history_trusted for diagnostics.
Practical note: the first OTA after installing this firmware may be a full upload because the stored manifest does not exist yet. This is expected. After that, differential uploads use the saved manifest.

v1.3.6 Addendum — reactive clean-pulse, XERAM 3G8B tube calibration, and UI updates

The periodic clean-pulse (fixed schedule every N cycles) was replaced with a reactive one: the sensor now stays at a single constant heater temperature (BME688_IAQ_HEATER_TEMP_C) all the time, and a higher 400°C pulse only fires after BME688_CLEAN_PULSE_SATURATION_STREAK (default 3) consecutive saturated readings (gas_adc ≥ 1023), not on a timer. Reason: the fixed schedule caused a periodic “tooth” in the IAQ graph at every pulse even though the pulse itself was excluded from history, because the sensor needs several cycles to thermally/chemically resettle afterwards.
The post-pulse exclusion window (BME688_CLEAN_PULSE_SETTLE_CYCLES) was increased from 4 to 8 cycles — real operating data showed the sensor's recovery after a pulse takes a variable 4 to 7 cycles, so a 4-cycle window let a smaller residual tooth slip through right at the edge. Since the pulse is now rare (only on genuine sustained saturation, not every ~20 minutes), the wider window costs far less than it did under the old fixed schedule.
The IAQ_TEXT classification was changed to: ≤100 Excellent, ≤200 Good, ≤250 Fair, ≤300 Poor, ≤400 Bad, >400 Very bad.
The Radiation tile and chart color thresholds were rescaled to more realistic values for the installed tube: green (safe) up to 1 µSv/h, orange (elevated) up to 10 µSv/h, red (high) above 10 µSv/h.
The actually installed GM tube was identified as XERAM 3G8B (the document and config.py had previously assumed J305/STS-5/SBM-20). From the tube's datasheet (“1 stroke/sec per mR/h”, i.e. 60 CPM per mR/h) the correct calibration constant was derived: CPM_PER_USV = 60 / 8.76 ≈ 6.85 (1 R = 8.76 mSv), replacing the previous value of 114 carried over from J305. The editable range of the CPM per 1 uSv/h field on the configuration page was widened from 50–200 to 1–200 so it would no longer clip the new value.
The web console (/console) had a bug where, during a brief device stall (e.g. while reading the sensor), the same log lines could appear duplicated when several concurrent requests to /api/console/recent overlapped. A guard against overlapping requests was added (the client will not start a new request until the previous one has completed).

In the OTA update log (/ota), the word “mtime” was replaced with “timestamp” in the change-reason messages — this is a cosmetic text change only; the comparison logic (size + CRC32, with mtime kept only as a historical fallback) is unchanged.

v1.3.7 Addendum — responsiveness and reliability improvements

The device's internal software was restructured so that slow operations (sensor readings, e-paper display refresh, sending data to InfluxDB) no longer freeze the web interface while they happen - the dashboard, configuration page and screenshot endpoint stay responsive at all times, even during a display refresh or a sensor read.
The dashboard pages now load their live data through a smaller, faster request, separate from the configuration page's much larger one.
Wi-Fi now remembers and prefers reconnecting to the same access point across restarts, but will automatically switch to a nearby access point of the same network if it offers a clearly stronger signal.
Wi-Fi power-saving mode is now disabled, which improves network responsiveness at a small cost in power consumption.
Fixed the browser clock and "next refresh" time occasionally showing about two hours ahead of the correct local time.
The one-time sensor warm-up heater pulse after startup is now skipped on a software restart or firmware update if the device was never actually unplugged, since the sensor is already warmed up.
The display now waits about two minutes after power-on before showing the first real sensor reading, avoiding a brief period where the BME688 air-quality sensor's very first reading is not yet representative.
Firmware version updated to v1.3.7.

Addendum v1.3.8 — protection against uploading the wrong firmware

Before every OTA update, the device now verifies that the uploaded package actually belongs to this product (the LK-29 weather/Geiger station), and not, for example, to the sibling LK-30 airQ product.
If the wrong package is uploaded by mistake, the update stops safely before any file is written, and the /ota page shows a clear error message about the target-device mismatch.
This change requires no user action during normal use - it only comes into play if the wrong installation package is selected by mistake.
The device's internal non-blocking architecture (see Addendum v1.3.7) was fully completed in this version.
Firmware version: v1.3.8.

Addendum v1.3.9 — smoother charts and more accurate temperature readings

Dashboard chart curves are now rendered smoothly without artificial "overshoot" on sharp changes in the measured values (e.g. radiation) - charts no longer look spiky.
The Y axis on every chart now always shows 5 equal parts, and the radiation chart shows a minimum range of 0-5 µSv/h so normal small fluctuations no longer look exaggerated.
Temperature, pressure and humidity measurements from the BME688 sensor are now taken before the sensor is heated for the air-quality (gas) measurement - this removes a small temperature bias caused by the chip's own heating.
Firmware version: v1.3.9.

Addendum v1.4.0 — chart averages and history protection across restarts

Each dashboard chart now shows its 24-hour average value ("x̄: ...") under the title.

When the device restarts (button, OTA update, or a Wi-Fi change), the measurement history is now saved and automatically restored after boot - so the charts no longer come back empty after a restart.
The /config page has a new button to permanently delete the saved history, in case you want to start fresh.
Firmware version: v1.4.0.