

FIELD NOTES FROM THE WORKBENCH

# BenchBook

## Kinetic Weather Instrument

Customer project handbook and development companion



### PROJECT 001

A customer handoff built around Michael Parks' original Mouser project: the files, the engineering context and a practical route to the next change.

**Read it. Rebuild it.  
Keep developing it.**

17 OCTOBER 2026  
EXAMPLE EDITION 0.3



THE COMPLETED INSTRUMENT / MOUSER ELECTRONICS [S1]

This edition includes the original article's images and links, a pinned copy of the GitHub firmware and EAGLE files, and GSG's proposed requirements, verification plan and AI development companion.

**EDITION STATUS** Historical source imported; engineering release NOT\_RELEASED. All engineering tests remain NOT\_RUN. Source discrepancies and missing mechanical files are documented for the next revision.

[Read the original Mouser article](#)



YOUR GUIDE TO THE INSTRUMENT

# Table of contents

Start with the project, explore the original build, then use the engineering records to plan and verify the next change. Each entry below is a clickable link.

01 / THE INSTRUMENT

|                                          |    |
|------------------------------------------|----|
| Project map                              | 03 |
| Original project and how to use this kit | 04 |

02 / THE ORIGINAL BUILD

|                                    |    |
|------------------------------------|----|
| Original circuit schematic         | 05 |
| Original breadboard prototype      | 06 |
| Original serial and cloud displays | 07 |
| Original airflow assembly          | 08 |
| Original ball standoff             | 09 |

03 / THE ENGINEERING BASELINE

|                                  |    |
|----------------------------------|----|
| Included GitHub source snapshot  | 10 |
| Source review before a new build | 11 |
| Hardware and interface record    | 12 |

04 / DEVELOPMENT AND HANDOFF

|                                      |    |
|--------------------------------------|----|
| Proposed customer acceptance         | 13 |
| Verification and integration plan    | 14 |
| Build operation and recovery         | 15 |
| Develop with an AI assistant         | 16 |
| Release inventory and open decisions | 17 |

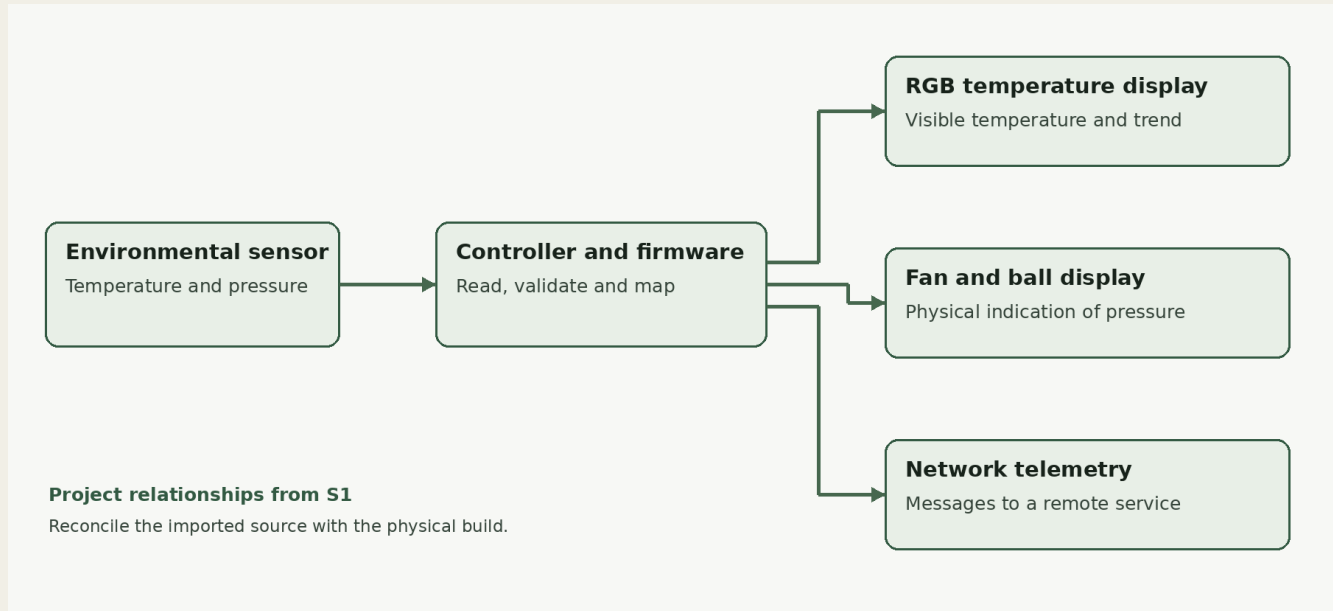
05 / THE REFERENCE SHELF

|                              |    |
|------------------------------|----|
| Terms and reference material | 18 |
|------------------------------|----|

## 01 / THE INSTRUMENT

# Project map

The published instrument combines an Arduino MKR WiFi 1010, BMP280 sensor, 8 × 8 RGB LED matrix, a blower-driven ball in a tube, and MQTT telemetry to Medium One. [S1] The physical outputs make the measurements visible, while the remote connection makes the data available elsewhere.



## What GSG brings together

The customer experiences one instrument. Its reliability depends on the interaction between the sensor, software, electrical drive, mechanics and network behavior. GSG's engineering scope should therefore define both the individual interfaces and the behavior of the complete system.

## A useful separation of responsibilities

For a new edition, separate acquisition and validation from rendering, fan control and networking. This lets the software exercise display and control behavior with recorded inputs before a physical test. Keep local sensing and displays independent of cloud connectivity. These are proposed design choices, not claims about the historical firmware.

## Physical behavior needs measurement

A fan command is an input to the mechanism. Ball height also depends on airflow, geometry, mass, friction and supply conditions. Characterize the assembled mechanism and record the calibration. Do not describe its height as closed-loop controlled unless a position sensor and feedback controller are actually present.

01 / THE INSTRUMENT

# Original project and how to use this kit

The physical thermometer and barometer provide the visual reference for the electronic instrument. The photographs in this edition come from the original Mouser article; GSG’s engineering records and proposed development process are additional handoff material.



Figure 1. The historical instruments pictured in the article. [S1] [Original image](#)

## A route through the package

| Start with                                | Then use                                                                    |
|-------------------------------------------|-----------------------------------------------------------------------------|
| The project map and original build images | Understand the sensing, display, airflow and telemetry paths.               |
| The source snapshot and review findings   | Locate the actual firmware and CAD; reconcile discrepancies before a build. |
| Requirements, verification and recovery   | Agree a change and record what was actually tested.                         |
| The AI companion and release inventory    | Continue development against a named, recoverable baseline.                 |



Original project: Michael Parks, P.E., for Mouser Electronics.

The complete local image collection and original link directory are also available in [article/index.html](#). The author portrait and four article social icons are retained with the source media.

## 02 / THE ORIGINAL BUILD

# Original circuit schematic

This is Figure 2 from the article. The native EAGLE schematic and board files are included with the pinned source snapshot. Keep this published drawing as a reference while establishing a verified as-built wiring record.

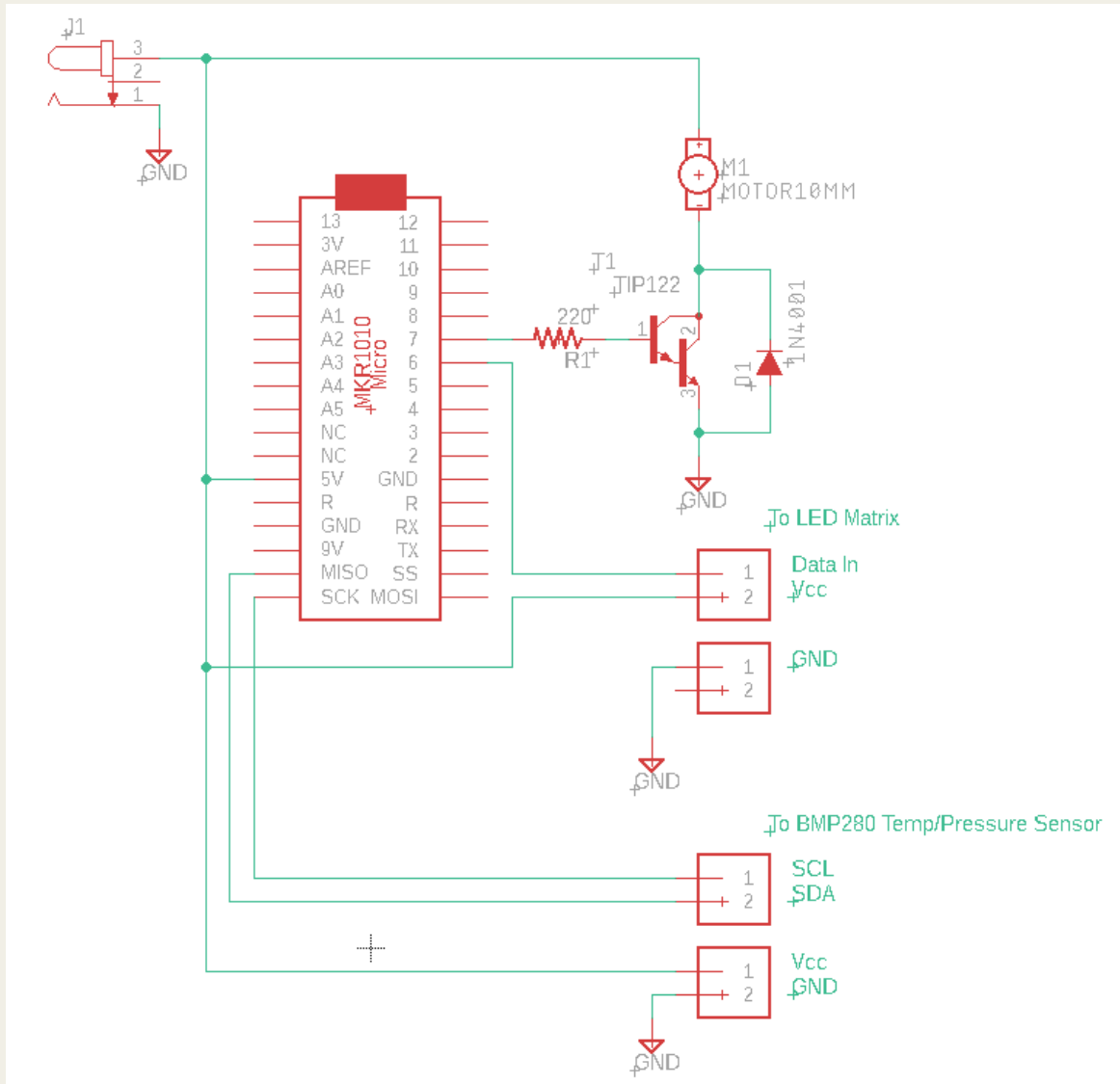


Figure 2. Original published schematic, preserved without image edits. [S1] [Original image](#)

## Reconcile before wiring

The sketch sets LEDPIN to 6 and FANPIN to 7. The article's fan-wiring narrative refers to D6. The EAGLE part named MKR1010 uses an ARDUINOMICRO deviceset, so its symbol is not sufficient proof of MKR WiFi 1010 physical pin assignments. Record the corrected, verified mapping before rebuilding. [S1, S5]

02 / THE ORIGINAL BUILD

# Original breadboard prototype

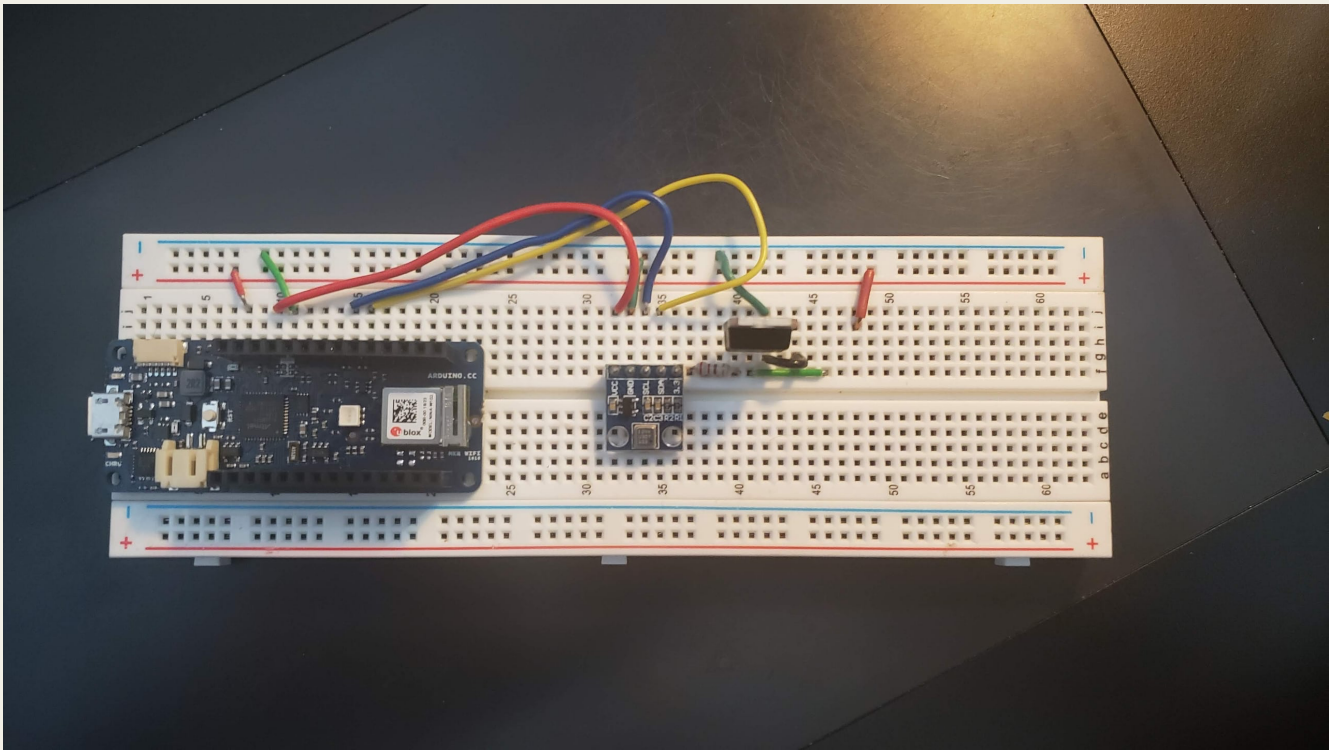


Figure 3. Controller, sensor and fan-driver circuitry on the original breadboard. [S1] [Original image](#)

## Turn the photograph into a build record

Use this photograph to understand the arrangement. Pair it with the source declarations and schematic, then record each actual connection, power rail and connector. A photograph is useful context, but it does not establish hidden connections or measured electrical limits.

| Included evidence                      | Next engineering record                                            |
|----------------------------------------|--------------------------------------------------------------------|
| Published prototype photograph         | Annotated as-built views and connector labels.                     |
| Firmware constants: LEDPIN=6; FANPIN=7 | Verified controller pin map and driver wiring.                     |
| EAGLE .sch and .brd files              | Reconciled schematic revision and checked board layout.            |
| Original Mouser shopping-cart link     | Reviewed BOM with current part numbers and approved substitutions. |

Parts reference: [Open the article’s original Mouser project cart](#)

## 02 / THE ORIGINAL BUILD

# Original serial and cloud displays

These are the article's original screenshots. They document the historical presentation and are not results from a new GSG test run.

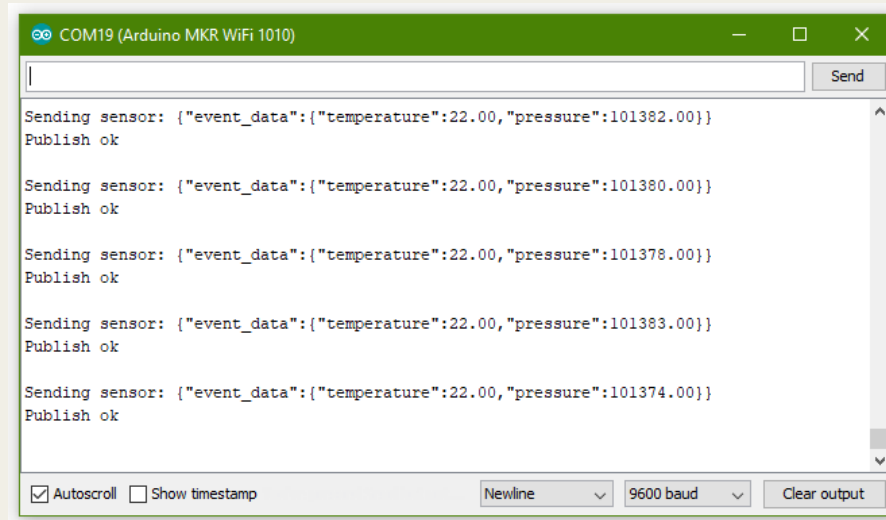


Figure 4. Serial diagnostic output from the original article. [S1] [Original image](#)

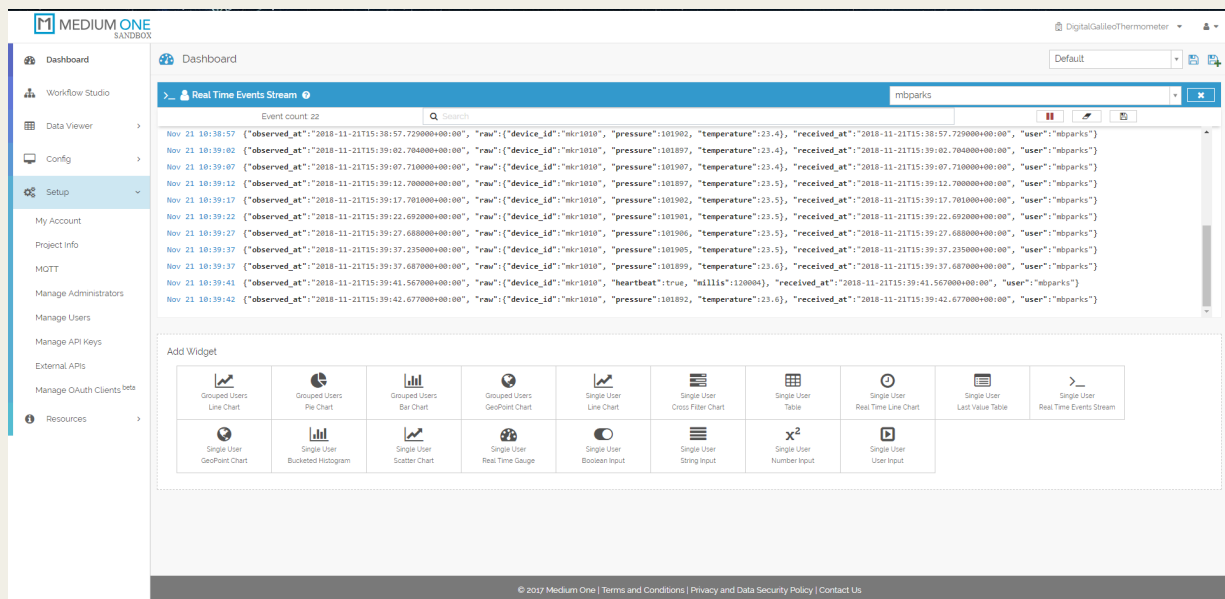


Figure 5. Historical Medium One dashboard from the original article. [S1] [Original image](#)

For a new release, capture fresh output with the firmware revision, units, device identity and configuration recorded. The source review explains the pressure-unit mismatch found in the sketch.

Original service references: [Mouser Medium One resources](#) • [Medium One](#)

## 02 / THE ORIGINAL BUILD

# Original airflow assembly



Figure 6. The tube photograph shows the vent holes and the ball. [S1] [Original image](#)



Figure 7. The outlet adapter at the base of the tube. [S1] [Original image](#)

Record tube dimensions, hole positions, adapter geometry and the assembled airflow response. The repository contains no STL files, so the photographs do not constitute a reproducible mechanical CAD handoff.

## 02 / THE ORIGINAL BUILD

# Original ball standoff

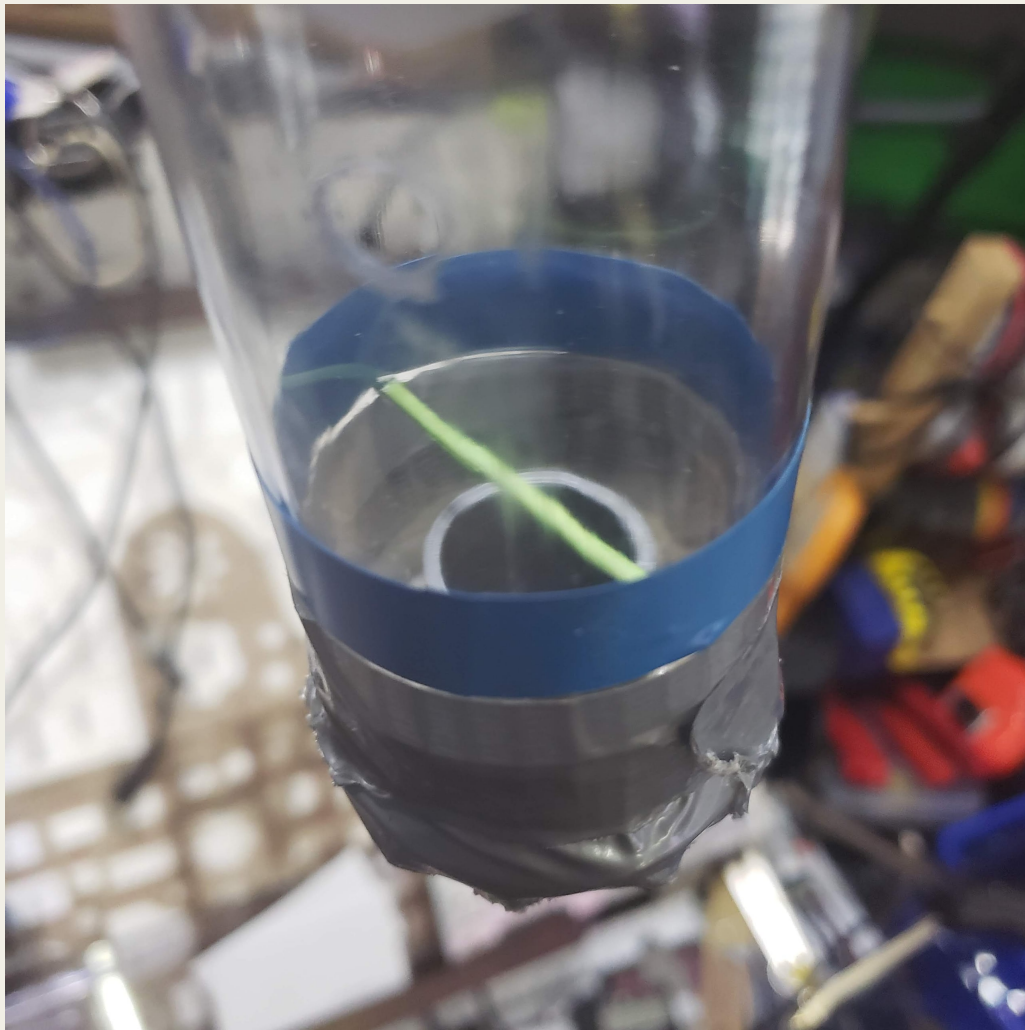


Figure 8. A crosspiece holds the ball above the outlet opening. [S1] [Original image](#)

## What the customer needs to retain

The mechanical record should include the ball specification, standoff position, vent pattern, retention features and fan calibration for the delivered assembly. Keep the original photograph as context and add dimensioned drawings or recovered CAD when available.

The existing T03 procedure provides a place to record ball travel, startup, settling and repeatability. No mechanical measurements or acceptance results have been added in this documentation revision.

## 03 / THE ENGINEERING BASELINE

# Included GitHub source snapshot

Repository: [Mouser-Electronics / MKR1010-Digital-Barometer](#)

Commit: 701459eacc0b7d0f0eb9bb2f4d49b8473687f662

Commit date: 25 July 2019. Retrieved: 17 October 2026. All nine repository files are included unchanged. Each file was checked against its Git blob hash; SHA-256 values are recorded in records/source\_snapshot.json.

| File inside source/MKR1010-Digital-Barometer                            | Purpose                                   |
|-------------------------------------------------------------------------|-------------------------------------------|
| Software/Firmware/MKR1010_DigitalBarometer/MKR1010_DigitalBarometer.ino | Original Arduino firmware.                |
| Software/Firmware/MKR1010_DigitalBarometer/yoursecretstuff.h            | Placeholder Wi-Fi and MQTT configuration. |
| Hardware/MKR1010_Barometer.sch                                          | Native EAGLE schematic.                   |
| Hardware/MKR1010_Barometer.brd                                          | Native EAGLE board file.                  |
| Documentation/Graphics/SchematicCapture.PNG                             | Repository schematic image.               |
| LICENSE and README.md                                                   | MIT license and upstream project notes.   |
| .gitattributes and .gitignore                                           | Original repository configuration.        |

## Source provenance and rights

The MIT license credits Mike Parks (2018). The original firmware header also retains its Adafruit attribution and BSD-license notice. Both are preserved. Article images are attributed separately to the original Mouser publication; they are not relabeled as MIT-licensed artwork.

## 03 / THE ENGINEERING BASELINE

# Source review before a new build

These findings come from static inspection of the included sketch and EAGLE schematic. The firmware has not been modified, compiled, flashed or physically tested for this edition. The detailed file and symbol references are in SOURCE-REVIEW.txt.

## Sensor driver differs from the article

The sketch includes Adafruit\_BMP085.h and creates Adafruit\_BMP085 bmp. Confirm the fitted sensor before selecting or replacing its driver.

## Configuration filename differs

The sketch includes secretstuff.h; the repository supplies yoursecretstuff.h. Create secretstuff.h in a working copy and enter local configuration there.

## Pressure units are inconsistent

readPressure() returns Pa, but pressureKPA is mapped against 85–110 without division by 1,000 or an explicit output clamp. Agree units and bounds before driving the fan.

## Startup depends on external connections

while (!Serial) and the Wi-Fi connection loop can block startup. An MQTT connection failure calls abort(). Independent local operation is proposed work.

## Pixel clearing is absent

The update sets pixels below the new integer temperature but does not clear pixels above it. Verify and correct decreasing-temperature behavior.

## Historical MQTT transport

The source uses WiFiClient and port 61619. Select and verify authentication, transport and provider behavior for a new release.

## Start in a working copy

Keep the imported folder immutable. Copy it to a development workspace, establish the actual sensor and board target, reconcile the pin map and record dependency versions. Resolve the findings in a separate revision; then run the software, hardware and integration checks tied to the requirements.

The current AI prompts and proposed requirements describe the intended next edition. They must not be read as capabilities already implemented by the archived firmware.

## 03 / THE ENGINEERING BASELINE

# Hardware and interface record

Populate this record from the released schematic, source code and the assembled device before rebuilding or modifying the instrument. The article establishes the concept; the release files establish its exact implementation.

| Interface             | Required record                                                                     | Verification                                                             |
|-----------------------|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------|
| Sensor to controller  | Fitted sensor, bus, address, pins, voltage and driver version.                      | Read identity where supported; exercise valid, failed and stale samples. |
| Controller to matrix  | Data pin, logic interface, supply, orientation, pixel order and brightness limit.   | Inspect signals and power; run reference pixel patterns.                 |
| Controller to fan     | Driver topology, control pin, PWM settings, supply, grounding and operating limits. | Measure commands and load behavior within the approved envelope.         |
| Fan to ball and tube  | Tube dimensions, vents, ball mass, retention, target bands and calibration.         | Record startup, settling, repeatability and containment.                 |
| Controller to service | Provider, transport, authentication, topic, payload, units and retry policy.        | Capture messages and exercise service and network outages.               |

## Resolve the sensor naming discrepancy

The article describes a BMP280 but refers to an Adafruit\_BMP085 firmware object. [S1] Adafruit documents a BMP280-specific driver. [S4] The included source confirms Adafruit\_BMP085.h and Adafruit\_BMP085 bmp. Check the fitted hardware before choosing a dependency. Record whether this is a documentation discrepancy or a real implementation difference.

## Document electrical limits explicitly

Arduino specifies 3.3 V I/O for the MKR WiFi 1010 and states that its pins are not 5 V tolerant. [S3] Confirm pull-ups, level interfaces, load supplies and the fan driver from their actual circuits and datasheets. Leave unresolved assignments blank instead of borrowing pin numbers from a different board.

## 04 / DEVELOPMENT AND HANDOFF

# Proposed customer acceptance

Agree these requirements before implementation. The numerical timing and cycling criteria below are starting proposals. Accuracy limits, mechanical bands and operating conditions remain decisions for GSG and the customer.

| ID  | Required behavior                                                   | Acceptance evidence                                                                                           |
|-----|---------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
| R01 | Valid temperature and pressure, with units and sample age.          | T01 Fixtures and comparison against agreed reference instruments.                                             |
| R02 | Approved temperature mapping and trend colors.                      | T02 Five temperatures plus rising, falling and steady sequences match reference frames.                       |
| R03 | Characterized fan output and bounded ball travel.                   | T03 Proposed three height bands, ten cycles and thirty-second holds; ball remains contained.                  |
| R04 | Local operation during network or service loss.                     | T04 Proposed five-minute outage with local updates no more than two seconds apart.                            |
| R05 | Invalid or stale data produces a visible fault and fan-off command. | T05 Proposed command within one second of invalid data; stale after five seconds. Measure rundown separately. |
| R06 | Explicit telemetry contract and protected credentials.              | T06 Decoded messages, reconnect checks, transport verification and redacted release files.                    |
| R07 | Repeatable firmware build and programming.                          | T07 A second environment reproduces the functional baseline from pinned source and dependencies.              |
| R08 | Recovery procedure and customer handoff.                            | T08 Proposed ten power cycles, one restore, smoke checks and an operating walkthrough.                        |

## What acceptance means

Each accepted requirement needs a result tied to a particular hardware revision, firmware revision, configuration and test setup. An approved exception must state its customer impact. Prototype acceptance covers the agreed use and conditions; production qualification is a separate scope.

## 04 / DEVELOPMENT AND HANDOFF

# Verification and integration plan

Run software checks first, then subsystem measurements and complete-system tests. The full procedures and evidence fields are in records/project\_record.json. Every test is currently NOT\_RUN.

| Stage             | Tests                                                                    | Record                                                                                 |
|-------------------|--------------------------------------------------------------------------|----------------------------------------------------------------------------------------|
| Software behavior | T01, T02, T05 and T06 with synthetic inputs.                             | Inputs, expected outputs, actual outputs, source revision and test command.            |
| Bench hardware    | Physical portions of T01 and T03, plus power and interface measurements. | Instrument identifiers, reference uncertainty, geometry, settings and measured values. |
| Complete system   | T02 through T06 on the assembled instrument.                             | Time-stamped logs, message captures, images or video and observed failure recovery.    |
| Release rehearsal | T07 and T08 with the delivered files.                                    | Clean build log, programming record, restore log and customer walkthrough.             |

## A test record customers can use

For every run, record the requirement and test IDs, operator, date, device serial, hardware revision, firmware commit and hash, configuration, equipment, environment, approved criteria, measured result, evidence filenames and remaining issues. Use records/test\_run\_template.json as the starting form.

## Example integration test

T04 begins with normal local operation and working telemetry. Disconnect Wi-Fi for five minutes, record the times of local updates, and restore connectivity. Repeat with the broker unavailable. Pass only when local behavior meets the agreed interval and reconnect behavior matches the telemetry contract. A network reconnect message alone does not prove that sensing continued.

## Keep evidence types distinct

A mock confirms behavior for the supplied inputs. A bench measurement characterizes a physical subsystem. An integration run checks interactions in the assembled device. A customer walkthrough confirms that the handoff is usable. Preserve those distinctions in the test report.

## 04 / DEVELOPMENT AND HANDOFF

# Build operation and recovery

A released BenchBook must let another engineer reproduce the software and let the customer restore a working device. This page defines that handoff procedure; the exact commands and files belong in the populated release manifest.

## Reproduce the release

1. Identify the device revision and obtain the matching schematic, BOM, pin map and mechanical information.
2. Restore the recorded source commit, board support package and exact dependency versions. Confirm the sensor driver against the actual device.
3. Create the local configuration from a redacted example. Enter credentials outside the source archive and record configuration version without recording secret values.
4. Build using the documented command. Retain the build log and binary hash; record any environment-dependent differences.
5. Program the approved target and perform the sensor, matrix, fan and telemetry smoke checks. Confirm that calibration and operating limits match that specific assembly.

## Operate the device

The final guide should identify the correct power connection, startup indication, normal display meanings, network-offline indication and sensor-fault indication. It should explain the local pressure interpretation and the difference between a visual indicator and a calibrated measurement. Include annotated photos of the delivered controls and connectors.

## Recover a known working version

Preserve the previous released binary and its configuration before modifying the device. The restore procedure must name the programming tool, connection, command, expected startup behavior and smoke checks. If a change affects fan calibration or stored settings, restore or migrate those values deliberately; restoring code alone may be insufficient.

## Resolve a problem systematically

If the ball behaves unexpectedly, compare the raw pressure, mapped command, driver output, supply and mechanical response in that order. Keep within the approved fan limits. If only the remote dashboard fails, inspect connectivity and message contracts while confirming local operation independently.

04 / DEVELOPMENT AND HANDOFF

# Develop with an AI assistant

Share PROJECT\_CONTEXT.txt, AI\_WORKING\_RULES.txt and the relevant records with your assistant, followed by SOURCE-REVIEW.txt and the included source files for the change. Keep secret values out. Ask it to identify the baseline, propose a bounded change and show the tests before modifying behavior.

## Worked change request

Add pressure trend to serial diagnostics and telemetry, while preserving the current local displays. The article identifies pressure-trend indication as a possible extension. [S1] This exercise gives the customer a concrete software task that can begin with deterministic data fixtures.

## Proposed behavior

Keep sixty consecutive one-minute pressure bins in pascals. Each bin holds the median of its valid samples. Compare the median of the latest five bins with the median of the earliest five. Report rising at a difference of at least +50 Pa, falling at at most -50 Pa, and steady between those thresholds. Report insufficient\_data until a complete window exists; a missing complete bin restarts warm-up.

The window and threshold are illustrative product choices. This feature describes a measured trend; it does not establish weather forecasting accuracy. Review memory use, sampling cadence and timer behavior on the actual controller before implementation.

| Fixture                                 | Expected result                               |
|-----------------------------------------|-----------------------------------------------|
| Fewer than sixty complete bins          | insufficient_data                             |
| Full window with +50 Pa difference      | rising                                        |
| Full window with -50 Pa difference      | falling                                       |
| Full window with 0 Pa difference        | steady                                        |
| A complete minute with no valid samples | Reset warm-up; insufficient_data              |
| Network loss while samples remain valid | Trend calculation and local displays continue |

## What to ask for

The complete prompt in context/PROMPTS.txt requests a patch plan, affected files, unit checks, regression coverage and rollback. Have GSG review hardware or interface changes and run the applicable physical tests. A plausible answer from an assistant is a proposal until its claims are checked.

## 04 / DEVELOPMENT AND HANDOFF

# Release inventory and open decisions

records/release\_manifest.json is the authoritative inventory for a delivered edition. It records file paths, hashes, device identity, revisions and acceptance. Its engineering release status is currently NOT\_RELEASED.

| Deliverable group         | Required in a customer release                                                              | This example                                                                                   |
|---------------------------|---------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| Project understanding     | Purpose, architecture, requirements and interface definitions.                              | Handbook and proposed records included.                                                        |
| Engineering files         | Source firmware, native schematic, readable schematic, BOM, pin map and mechanical files.   | Historical firmware, EAGLE and schematic image included; reconciliation and STL files pending. |
| Reproduction and recovery | Pinned dependencies, commands, release binary, configuration example and restore procedure. | Placeholder configuration included; pinned tools, binary and recovery evidence pending.        |
| Verification evidence     | Test results, calibration, reference conditions, known issues and acceptance record.        | Procedures included; all tests NOT_RUN.                                                        |
| Development companion     | AI context, working rules and bounded example prompts.                                      | Included as editable text.                                                                     |
| Continuing ownership      | File rights and licenses, support scope and provider/account responsibilities.              | Upstream MIT license and source attribution included; engagement terms remain open.            |

## Decisions that unlock the next edition

First reconcile the source and sensor driver. Then establish the pin map, electrical limits, mechanics and a repeatable firmware build. Agree operating ranges, measurement tolerances, display behavior and fan calibration. Select and verify the telemetry service. These decisions turn the example into an executable engineering plan.

## Customer acceptance record

The delivered edition should identify the accepted configuration, applicable test results, agreed exceptions, customer representative, acceptance date and support period. Keep later customer modifications as a new revision so the tested baseline remains recoverable.

## 05 / THE REFERENCE SHELF

# Terms and reference material

| Term                        | Meaning in this BenchBook                                                                                                   |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| BOM                         | Bill of materials: the parts, quantities and exact identifiers used in the build.                                           |
| Firmware                    | Software that runs on the instrument's controller.                                                                          |
| Interface                   | An agreed electrical, mechanical or software connection between parts of the system.                                        |
| PWM                         | Pulse-width modulation: a time-varying control signal whose duty cycle can command a driver.                                |
| MQTT                        | A messaging protocol used for telemetry between the device and a service.                                                   |
| Regression test             | A check that existing behavior still works after a change.                                                                  |
| Release manifest            | The record tying a delivered version to its files, revisions and checksums.                                                 |
| Verification and validation | Verification checks agreed requirements. Validation checks whether the delivered system serves the customer's intended use. |

## References

S1 Mouser Electronics. A Modern Take on Galileo's Thermometer and Goethe's Barometer.

<https://www.mouser.com/en/applications/modern-take-galileos-thermometer-geothses-barometer/>

S2 Arduino. MKR WiFi 1010 documentation and datasheet.

<https://docs.arduino.cc/hardware/mkr-wifi-1010/>

S3 Adafruit. BMP280 breakout Arduino test.

<https://learn.adafruit.com/adafruit-bmp280-barometric-pressure-plus-temperature-sensor-breakout/arduino-test>

S4 Mouser Electronics. Pinned original source repository.

<https://github.com/Mouser-Electronics/MKR1010-Digital-Barometer/tree/701459eacc0b7d0f0eb9bb2f4d49b8473687f662>

S5 Adafruit. BMP085 example: readPressure() reports Pa.

<https://github.com/adafruit/Adafruit-BMP085-Library/blob/master/examples/BMP085test/BMP085test.ino>

Reviewed 17 October 2026. The Mouser article supplies the historical project context. Vibay supplies the public reference for an AI development guide. Arduino and Adafruit supply component documentation. Proposed requirements, tests and release practices in this example are GSG service-design recommendations.