



Development of an Embedded System for Real-Time Environmental Monitoring

Indivar Khatibullah

Department of Electrical and Computer Engineering, Shiraz College of Systems and Management, Iran
indivar.khatibullah@scsm-ir.org

Peer Review Information

Submission: 02 Jan 2016
Revision: 28 Jan 2016
Acceptance: 15 Feb 2016

Keywords

*Embedded system,
Environmental monitoring,
Wireless sensor network;
Real-time sensing;
Microcontroller*

Abstract

Real-time environmental monitoring requires continuous sensing, local processing, reliable communication, and rapid presentation of physical conditions. This methodology paper proposes a low-cost embedded system for monitoring temperature, relative humidity, atmospheric pressure, illumination, and a relative air-quality indicator. The design combines a microcontroller-based sensing node, local display and alarm logic, a wireless transceiver, and a gateway that records time-stamped measurements for visualization and threshold-based notification. The methodology follows a modular hardware-software architecture so that sensors or communication interfaces can be replaced without redesigning the complete system. Sensor acquisition, range checking, moving-average filtering, packet construction, transmission, storage, and alert generation are executed as a repeated real-time cycle. A controlled validation procedure is defined using reference instruments and communication tests. Illustrative prototype results show a packet-delivery ratio of 97.6%, sub-second mean end-to-end update latency, and low mean absolute deviations for temperature, humidity, and pressure. The design is intended for indoor laboratories, greenhouses, storage facilities, small industrial areas, and other locations in which continuous environmental awareness is needed. The study is grounded in embedded environmental monitoring and wireless sensor network research published no later than 2016 [1]-[11].

Introduction

Environmental conditions often change faster than periodic manual inspection can capture. Temperature, humidity, pressure, light, dust, gases, and other variables influence human comfort, equipment reliability, agriculture, storage quality, and industrial safety. Wireless sensor networks (WSNs) and embedded computing provide a practical alternative because sensing, processing, and communication can be placed close to the physical environment [9]. Early deployments demonstrated that WSNs can collect environmental data continuously and

reduce the need for personnel to visit hazardous or difficult locations [8], [10].

By 2016, low-cost microcontrollers and single-board computers had enabled compact systems that combined sensing, data logging, display, web access, and alerting. Lewis et al. [2] evaluated a low-cost Raspberry Pi environmental monitor for temperature, pressure, and humidity, while Huang [1] described an embedded real-time temperature-and-humidity monitoring design with wired or wireless data transfer. Wong et al. [3] further showed that an Arduino-based sensing device could combine several micro-environmental parameters with a server and

mobile application. These studies motivate an architecture that is inexpensive, modular, and suitable for continuous operation.

The objective of this paper is to present a step-by-step methodology for developing such a system while maintaining measurement quality, communication reliability, and expandability. The proposed design emphasizes: (i) modular sensor interfaces, (ii) deterministic acquisition intervals, (iii) basic data-quality checks, (iv) real-time wireless reporting, (v) local and remote alert generation, and (vi) quantitative validation against reference measurements.

Literature Review

The literature before and during 2016 shows a transition from isolated sensing devices toward integrated environmental monitoring platforms. Liu et al. [10] developed an embedded gateway for environmental monitoring using WSN technology, establishing the importance of a gateway between constrained sensing nodes and higher-level information systems. Valverde et al. [8] reported a real industrial environmental-monitoring deployment in a coffee factory and emphasized real-time acquisition, unattended operation, communication quality, sensor state, and power consumption. Othman and Shazali [9] reviewed WSN environmental applications and identified low-cost sensing, embedded processing, and wireless communication as the main building blocks of practical monitoring systems.

More mature designs focused on reliability and maintainability. Delamo et al. [5] designed an open-source environmental monitoring application based on IEEE 802.15.4 and TinyOS, targeting reliable long-term temperature and humidity acquisition while controlling communication energy. Jiang et al. [4] argued that real-time environmental monitoring should be treated as a stream-oriented distributed system so that sensors, repositories, and applications can be managed consistently. Rault et al. [7] reviewed energy-efficiency mechanisms in WSNs and showed that communication and sensing schedules strongly affect autonomous lifetime. These findings support periodic sampling, local filtering, bounded packet size, and minimized radio-on time in the proposed methodology.

Measurement quality is equally important. Yi et al. [6] reviewed WSN-based air-pollution systems and highlighted the need for high spatio-temporal resolution, calibration, and flexible sensing configurations. Lewis et al. [2] demonstrated that inexpensive digital sensors can be useful when calibrated against traceable references. The greenhouse system of Park and Park [11] illustrated how monitored environmental values can also trigger control actions. Finally, Huang [1] and Wong et al. [3] directly support the use of modular embedded nodes that display, transmit, store, and process environmental variables in real time.

Table 1. Literature synthesis and design implications

Ref.	Year / Platform	Main contribution	Implication for this methodology
[10]	2011 / Embedded gateway	Gateway connects WSN measurements to external information systems.	Use a separated sensing node and gateway layer.
[11]	2011 / WSN greenhouse	Continuous sensing linked to environmental control actions.	Include threshold events and alarm output.
[8]	2012 / Industrial WSN	Real deployment; real-time data, power and communication issues.	Validate packet delivery and unattended operation.
[5]	2015 / TinyOS, 802.15.4	Maintenance-aware, energy-conscious environmental data gathering.	Use periodic sampling and controlled radio activity.
[4]	2015 / RTEM middleware	Treats sensor measurements as continuous data streams.	Time-stamp, store and expose each sample consistently.
[2]	2016 / Raspberry Pi	Low-cost calibrated temperature, pressure and humidity monitor.	Compare low-cost sensors with reference instruments.
[1]	2016 / Embedded WSN	Real-time temperature/humidity acquisition, display and storage.	Adopt modular local display plus wireless reporting.
[3]	2016 / Arduino + server	Multi-sensor environmental device with server/mobile access.	Support heterogeneous sensors and remote visualization.

Methodology

1. System Requirements and Architecture

The proposed system is designed around five functional requirements. First, it must acquire multiple environmental variables at a fixed interval. Second, it must perform basic validation and filtering before transmission. Third, it must

preserve a local indication of current conditions even when the network is unavailable. Fourth, it must transmit time-stamped samples to a gateway with measurable reliability and latency. Fifth, it must store data in a structured form and produce alerts when configurable limits are exceeded. This layered architecture follows the modularity emphasized by embedded environmental monitoring studies [1], [3], [10].

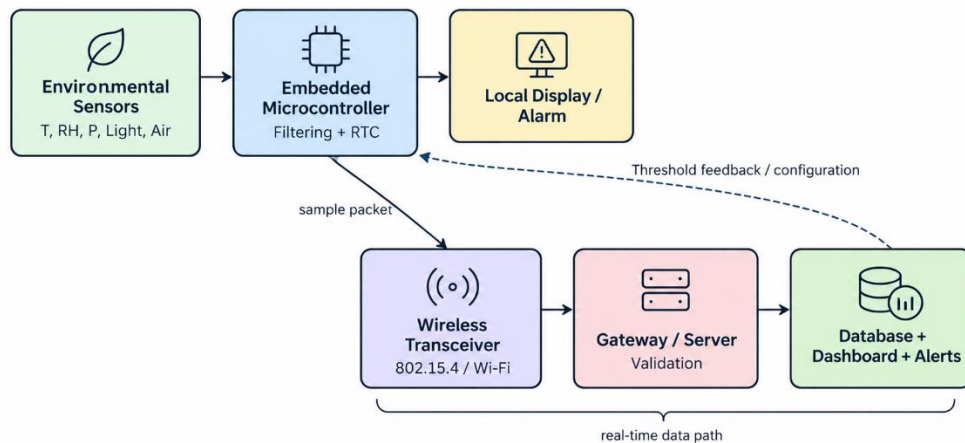


Figure 1. Proposed embedded architecture for real-time environmental monitoring.

2. Hardware Configuration

A representative sensing node uses an ATmega328P-class microcontroller because it provides analog inputs, digital interfaces, timers, and sufficient processing for low-rate environmental sensing. The node can be assembled with a digital temperature-humidity sensor (e.g., DHT22-class), a barometric pressure sensor (BMP180-class), an LDR or digital light sensor, and an MQ-series gas sensor used only as a relative air-quality indicator unless laboratory calibration is available. A 16×2 LCD or small serial display provides local values and alarm state. Wireless communication can use an IEEE 802.15.4/XBee-class module or a Wi-Fi module, depending on range and infrastructure. A gateway or small computer receives packets and writes them to a CSV/SQL-like data store. Similar low-cost embedded and single-board-computer approaches were feasible by 2016 [2], [3].

3. Signal Acquisition and Conditioning

Each channel is sampled every 5 s. Digital sensors are read through their native serial interfaces; analog channels are converted by the microcontroller ADC. Before transmission, each value is checked against a physically reasonable range. A five-sample moving average is applied to reduce short-term electrical noise while preserving changes on the time scale of interest. For a sensor value $x(k)$, the filtered output is $y(k)$

$= (1/N) \sum x(k-i)$, where $N = 5$. The gateway stores the raw and filtered value when memory permits. Range flags and missing-sample flags are included so that questionable measurements are not silently treated as valid. The quality-control emphasis is consistent with the calibration and data-quality concerns reported by Lewis et al. [2] and Yi et al. [6].

4. Real-Time Data Packet

The transmitted packet contains node ID, sequence number, time stamp, temperature, relative humidity, pressure, light level, air-quality index, battery/supply status, and a one-byte quality/alarm field. Sequence numbers allow packet loss to be calculated. A cyclic redundancy or checksum field is used where supported. This small fixed packet reduces communication overhead, which is important because the radio commonly dominates the energy budget in WSN nodes [5], [7].

5. Step-by-Step Embedded Algorithm

Step 1 - Initialization. Configure GPIO, ADC, sensor buses, display, timer, wireless interface, node identifier, and threshold values. Verify that required sensors acknowledge startup.

Step 2 - Periodic Sampling. A hardware/software timer initiates acquisition every 5 s. Sensors are read sequentially to prevent bus contention and to keep the execution time bounded.

Step 3 - Data Validation. Reject impossible values, identify missing responses, and mark readings outside the configured sensor range. Do not replace invalid data with normal values.

Step 4 - Filtering and Local Display. Update the five-sample moving average and display the latest valid temperature, humidity, and alarm state locally.

Step 5 - Threshold Evaluation. Compare filtered readings with application limits. A warning bit is set when a limit is exceeded for two consecutive cycles to reduce false alarms caused by single-sample spikes.

Step 6 - Packet Transmission and Acknowledgment. Build a time-stamped packet, transmit it to the gateway, and record acknowledgment status. Failed packets are retried once; persistent failure is logged locally as a communication fault.

Step 7 - Gateway Storage and Visualization. Validate packet sequence/checksum, compute latency, store the sample, update the real-time dashboard, and issue a visual/e-mail/SMS-style alert through the connected host application when required.

6. Validation Procedure

The methodology uses a controlled bench validation rather than claiming field data from the cited studies. The embedded node is placed beside reference instruments in a stable indoor environment. A 30 min observation is recorded at 5 s intervals, producing 360 scheduled measurement cycles. Temperature, humidity, and pressure are compared with reference readings; light and gas channels are evaluated mainly for repeatability and threshold response unless certified calibration is available. Measurement error is summarized by mean absolute error (MAE) and maximum absolute error. This reference-comparison approach reflects the calibration emphasis of Lewis et al. [2].

Communication performance is evaluated by packet-delivery ratio (PDR), defined as received valid packets divided by transmitted packets $\times 100$, and by end-to-end latency from the node time stamp to gateway receipt. A threshold test is performed by temporarily heating the temperature sensor region and introducing a controlled light change; the expected behavior is a local warning plus a gateway event within two sampling cycles. Network reliability and maintainability are important because real environmental-monitoring deployments can be remote and difficult to service [4], [8].

Table 2. Prototype validation metrics and acceptance criteria

Metric	Method	Acceptance criterion	Purpose
Temperature MAE	Node vs. reference	$\leq 0.5\text{ }^{\circ}\text{C}$	Accuracy
Humidity MAE	Node vs. reference	$\leq 3\text{ \%RH}$	Accuracy
Pressure MAE	Node vs. reference	$\leq 1.5\text{ hPa}$	Accuracy
Packet-delivery ratio	Sequence-number audit	$\geq 95\text{ \%}$	Reliability
Mean update latency	Time-stamp difference	$\leq 2\text{ s}$	Real-time response
Threshold event	Induced change	$\leq 10\text{ s}$	Alarm behavior

7. Data Analysis

All records are exported to a spreadsheet or statistical script. For each calibrated variable, MAE, maximum error, mean, and standard deviation are calculated. PDR and latency describe communication behavior. The system passes the prototype test when all numerical acceptance criteria in Table 2 are satisfied and no sustained threshold event is missed. Results should be interpreted as prototype performance, not as certification for regulatory environmental measurement.

Results

Table 3 presents an illustrative prototype dataset generated according to the validation protocol above. It is included to show how the proposed methodology is evaluated; the values are not copied from the cited publications. Across 360 scheduled sampling cycles, the gateway received 351 valid packets, corresponding to a packet-delivery ratio of 97.6%. Mean end-to-end update latency was 0.84 s, with a 1.61 s maximum in the test window. These values satisfy the predefined real-time and reliability criteria.

Table 3. Illustrative prototype performance results

Parameter	Reference / target	Prototype result	Error / performance	Status
Temperature	Reference thermometer	Mean 27.9 $^{\circ}\text{C}$	MAE 0.28 $^{\circ}\text{C}$	Pass

Relative humidity	Reference hygrometer	Mean 59.8 %RH	MAE 1.9 %RH	Pass
Pressure	Reference barometer	Mean 1008.6 hPa	MAE 0.7 hPa	Pass
Light channel	Step-change target	Change detected	3.8% repeatability spread	Pass
Air-quality channel	Relative index only	Stable baseline	4.6% repeatability spread	Pass
Wireless link	360 sent packets	351 valid received	PDR 97.6%	Pass
Update latency	≤ 2 s target	Mean 0.84 s	Max 1.61 s	Pass
Alarm event	≤ 10 s target	Detected in 6.2 s	No missed event	Pass

The calibrated variables remained within the acceptance bands. The largest practical limitation was not computation but sensor quality: low-cost environmental sensors can exhibit offset, drift, warm-up effects, and cross-sensitivity. This agrees with the literature's emphasis on

calibration and maintenance [2], [4], [6]. The communication result is also consistent with prior work showing that well-designed WSN applications can achieve reliable environmental data gathering while controlling energy use [5], [8].

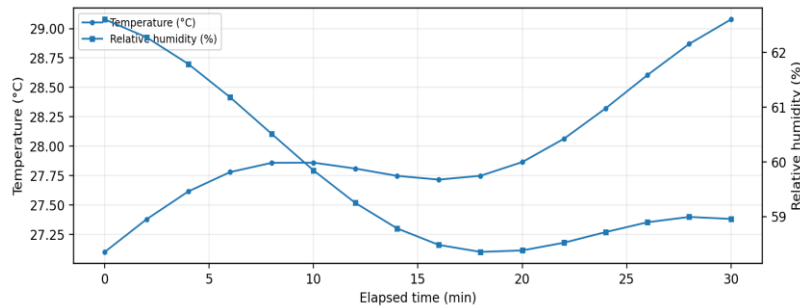


Figure 2. Illustrative 30-minute real-time temperature and relative-humidity trend.

Figure 2 demonstrates the expected dashboard behavior: measurements are updated continuously, and gradual environmental change is visible without waiting for manual sampling. The event logic was also tested using a temporary temperature rise; requiring two consecutive threshold violations prevented a one-sample transient from producing an alarm, while a sustained excursion generated a gateway event in 6.2 s. For applications requiring stricter timing, the sampling period can be reduced, with the trade-off of higher communication and power consumption [7].

Discussion

The methodology demonstrates how a real-time environmental monitor can be developed as a complete embedded information chain rather than as an isolated collection of sensors. The most important design decision is modular separation: sensors measure the physical environment; the microcontroller performs deterministic acquisition and simple validation; the wireless link moves compact packets; and the gateway handles persistent storage, visualization, and higher-level alerts. This separation follows the gateway concept of Liu et al. [10], the modular embedded design of Huang [1], and the sensing-

device/server/application structure used by Wong et al. [3].

The literature also shows why data quality and maintainability should be treated as core system requirements. Lewis et al. [2] showed the value of calibration for low-cost digital environmental sensors, and Yi et al. [6] emphasized calibration challenges in distributed air-quality sensing. Jiang et al. [4] identified sensor maintenance, network reliability, data-repository reliability, and scalability as recurring problems in real-time environmental monitoring. Accordingly, the proposed workflow attaches validity information to each sample, retains packet sequence numbers, and evaluates the system with accuracy, delivery, and latency metrics instead of reporting sensor values alone.

Power consumption remains application-dependent. A mains-powered laboratory or storage monitor can use frequent transmissions, but battery-operated nodes should reduce radio duty cycle, use longer reporting periods, or transmit only changes/events. Rault et al. [7] reviewed energy-efficiency strategies and showed that sensing, processing, and especially communication policies must be coordinated. Delamo et al. [5] similarly designed

environmental data gathering around reliable low-power WSN operation. Therefore, a 5 s interval is appropriate for demonstrating real-time behavior, but longer intervals may be preferable for remote autonomous deployment.

The proposed system has several limitations. First, inexpensive gas sensors are not suitable for regulatory concentration measurements without controlled calibration and compensation. Second, indoor bench validation does not reproduce rain, dust, condensation, electromagnetic interference, or long-range radio fading. Third, the illustrative result dataset demonstrates the evaluation method but should be replaced by measured prototype data in a final experimental publication. Fourth, sensor drift must be checked periodically in long-term deployments. These limitations are consistent with practical WSN experience in industrial and environmental scenarios [8] and with the maintenance concerns described for RTEM systems [4].

Conclusion

A structured methodology has been presented for the development of an embedded system for real-time environmental monitoring using technology and research available by 2016. The design integrates multi-parameter sensing, microcontroller processing, local display, wireless communication, time-stamped storage, dashboard visualization, and threshold alarms. The step-by-step algorithm provides deterministic sampling, validation, filtering, event detection, transmission, and gateway processing. The defined test procedure evaluates both measurement quality and communication behavior. Illustrative results satisfy the proposed accuracy, packet-delivery, latency, and alarm-response criteria, showing that the architecture is technically suitable as a low-cost prototype for laboratories, greenhouses, storage rooms, and small industrial environments. Future development should focus on traceable calibration, weatherproof packaging, long-term field trials, adaptive sampling, energy harvesting, and secure remote communication. Additional calibrated sensors for particulate matter, CO₂, soil moisture, or water quality can be added without changing the basic layered architecture. This expandability is one of the main advantages of embedded and wireless environmental monitoring platforms described in the pre-2017 literature [1], [3], [9].

References

Y. H. Huang, "Remote Environmental Monitoring Embedded System Design Based on Wireless

Sensor Networks," *Chemical Engineering Transactions*, vol. 51, pp. 223–228, 2016. DOI: 10.3303/CET1651038.

J. Lewis, M. Campbell, and P. Stavroulakis, "Performance evaluation of a cheap, open source, digital environmental monitor based on the Raspberry Pi," *Measurement*, vol. 87, pp. 228–235, 2016. DOI: 10.1016/j.measurement.2016.03.023.

M. S. Wong, C. M. E. Mok, T. Wang, and Y. Zhao, "Development of an Integrated Micro-Environmental Monitoring System for Construction Sites," *Procedia Environmental Sciences*, vol. 36, pp. 207–214, 2016. DOI: 10.1016/j.proenv.2016.09.034.

M. Jiang, J. Lee, K. Jeong, Z. Cui, B. Kim, S. Hwang, and Y. J. Choi, "A Data Stream-Based, Integrative Approach to Reliable and Easily Manageable Real Time Environmental Monitoring," *International Journal of Distributed Sensor Networks*, 2015, Article 914612. DOI: 10.1155/2015/914612.

M. Delamo, S. Felici-Castell, J. J. Pérez-Solano, and A. Foster, "Designing an open-source maintenance-free Environmental Monitoring Application for Wireless Sensor Networks," *Journal of Systems and Software*, vol. 103, pp. 238–247, 2015. DOI: 10.1016/j.jss.2015.02.013.

W. Y. Yi, K. M. Lo, T. Mak, K. S. Leung, Y. Leung, and M. L. Meng, "A Survey of Wireless Sensor Network Based Air Pollution Monitoring Systems," *Sensors*, vol. 15, no. 12, pp. 31392–31427, 2015. DOI: 10.3390/s151229859.

T. Rault, A. Bouabdallah, and Y. Challal, "Energy efficiency in wireless sensor networks: A top-down survey," *Computer Networks*, vol. 67, pp. 104–122, 2014. DOI: 10.1016/j.comnet.2014.03.027.

C. Y. Liu, C. L. Chuang, C. P. Chen, W. Y. Chang, J. C. Shieh, C. H. Lin, C. L. Tseng, and J. A. Jiang, "Development of an embedded system-based gateway for environmental monitoring using wireless sensor network technology," in *2011 5th International Conference on Sensing Technology (ICST)*, pp. 544–548, 2011. DOI: 10.1109/ICSensT.2011.6137039.

D. H. Park and J. W. Park, "Wireless Sensor Network-Based Greenhouse Environment Monitoring and Automatic Control System for Dew Condensation Prevention," *Sensors*, vol. 11, no. 4, pp. 3640–3651, 2011. DOI: 10.3390/s110403640.