



Design and Implementation of Microcontroller-Based Automatic Energy Meter Reading Systems

Thabo Kalimuthu

Department of Computer Science and Engineering, Kavir Polytechnic University of Technology, Iran
thabo.kalimuthu@kput-ir.net

Peer Review Information

Submission: 18 Jan 2017

Revision: 11 Feb 2017

Acceptance: 16 March 2017

Keywords

*Automatic Meter Reading,
Microcontroller, Energy
Meter, GSM, Smart Meter.*

Abstract

Automatic meter reading (AMR) removes repeated manual visits to electricity meters and creates a direct path from measured energy consumption to billing and utility monitoring. This methodology paper presents a low-cost microcontroller-based AMR architecture in which the pulse output of a single-phase electronic energy meter is electrically isolated, counted by an ATmega-class microcontroller, converted to kilowatt-hours, displayed locally, stored in non-volatile memory, and transmitted through a GSM module at scheduled intervals or on request. The method emphasizes modular hardware, interrupt-driven pulse acquisition, data integrity, simple tariff calculation, and communication retry logic. The proposed validation uses a 1000 imp/kWh pulse constant and a controlled numerical pulse-generation model so that measurement error can be quantified without claiming unperformed laboratory measurements. Across six representative load-duration cases, the pulse-counting method produces a maximum quantization error of 0.44% and a mean absolute percentage error of approximately 0.12%. The architecture therefore demonstrates how a modest microcontroller can automate energy acquisition and billing while preserving local display and storage functions. The design is intended as a reproducible foundation for academic prototypes and can be extended with tamper detection, prepaid operation, demand-side control, or advanced metering communication.

Introduction

Electricity billing traditionally depends on periodic physical inspection of consumer meters. Although this approach is simple, it introduces delays, transcription errors, labour cost, inaccessible-meter problems, and limited visibility of consumption between billing cycles. The migration from electromechanical meters toward electronic meters created a practical opportunity to automate this process because electronic meters can expose pulse outputs or digital interfaces that represent accumulated energy. Early wireless energy-meter studies demonstrated that readings could be obtained with minimal human intervention, first through

short-range wireless links and later through embedded communication networks [1].

The essential idea of AMR is to acquire the meter value automatically and transfer it to a utility-side system without requiring a person to visit each premises. Embedded metering concepts further showed that a local processor could perform not only measurement but also billing, demand supervision, and connection control [2]. Likewise, ARM-based power-management systems integrated energy acquisition with GPRS communication and remote control, illustrating the importance of the microcontroller as the local decision-making unit [3]. These developments established the technical basis for a compact

system containing a meter interface, processor, memory, display, communication module, and optional relay output.

Communication choice strongly affects AMR coverage, cost, and complexity. Wireless mesh approaches using ZigBee provided neighbourhood-scale data collection [4], while GSM and related cellular systems offered the advantage of using an already deployed public network. Wireless automated digital meters also showed how telecommunication could reduce the manual steps involved in reading and billing [5]. Later GSM-based designs combined meter acquisition, local display, instant billing, and SMS transfer [6], making cellular AMR attractive where fixed communication infrastructure is unavailable.

This paper develops a methodology for a microcontroller-based automatic energy meter reading system intended for a single-phase consumer installation. The principal objectives are to: (i) capture meter pulses reliably; (ii) convert pulses into energy units; (iii) maintain cumulative readings in non-volatile memory; (iv) display consumption and billing information locally; (v) transmit readings through GSM; and (vi) evaluate the expected measurement accuracy of the pulse-counting method using a transparent numerical validation. The scope is limited to methodology and prototype-level design; it does not claim field certification, utility-grade accuracy-class testing, or a completed commercial deployment.

Literature Review

Koay et al. [1] presented a Bluetooth-enabled energy meter and demonstrated automatic polling as a way to reduce human involvement in meter reading. Although Bluetooth offers low-cost local connectivity, its short range makes it less suitable for direct utility-wide access without additional gateways. Maitra [2] moved the concept toward an embedded meter with local billing and demand functions, showing that the processor can become part of the metering logic rather than serving only as a communication interface.

In 2008, Li et al. [3] used an ARM microprocessor and GPRS for remote power management and meter reading. Their work is directly relevant to the present methodology because it links an electronic kWh meter, embedded terminal, and management centre. In the same period, Cao et al. [4] proposed a mesh-network AMR system built around meter nodes, a data collector, and a management server. Bharath et al. [5] also demonstrated a wireless automated digital energy meter for remote transfer of readings. Together, these studies show two recurring

architectural patterns: a local embedded acquisition unit and a communication path to a remote billing or management point.

Sehgal et al. [6] integrated GSM with an electronic meter for instant billing, including local display and remote access to consumption information. Khalifa, Naik, and Nayak [7] later surveyed AMR communication protocols and highlighted reliability, scalability, time constraints, and secure communication as central design issues. This supports the use of buffered local storage and retry logic in the present methodology rather than assuming every transmission will succeed immediately.

Landi et al. [8] demonstrated an ARM-based energy management system with smart-meter data and a web server, emphasizing that local measurement becomes more useful when combined with accessible consumption information. Ashna and George [9] subsequently proposed GSM-based AMR with instant billing and a web interface, reinforcing the suitability of cellular communication for low-cost remote meter retrieval. By 2017, Labib et al. [10] had implemented a low-cost smart energy meter using a microcontroller, current and potential transformers, local computation, and cellular communication, with additional remote configuration and protection capabilities.

The literature therefore indicates a clear progression from local wireless reading to networked embedded metering, but a useful academic prototype still benefits from a deliberately simple architecture. The present methodology adopts pulse counting from an existing electronic meter rather than re-implementing the entire metrology front end. This reduces analog-design complexity and allows the study to concentrate on reliable acquisition, computation, storage, display, and remote reading.

Methodology

1. System Architecture

The proposed system is organized into five functional stages: energy-meter pulse generation, electrical isolation and signal conditioning, microcontroller-based counting and computation, local display/storage, and GSM-based transmission. The electronic meter is assumed to provide a pulse output proportional to consumed energy. For a meter constant K of 1000 impulses per kilowatt-hour, one pulse corresponds to 0.001 kWh. An optocoupler isolates the meter from the low-voltage digital electronics and presents a clean logic transition to the microcontroller interrupt input. The processor increments a persistent pulse counter, converts the accumulated count to energy,

calculates a billing estimate, refreshes the LCD, and periodically stores the latest value in EEPROM.

2. Hardware Components

A practical prototype can be built around an ATmega328P-class 8-bit microcontroller or a comparable device with external interrupt capability, UART communication, timers, and

non-volatile memory. The design does not depend on a specific board form factor. The GSM interface may use a SIM800/SIM900-class modem controlled by AT commands over UART. A 16×2 LCD provides local visibility, while a regulated supply powers the logic and communication blocks. Table 1 summarizes the principal hardware elements and their roles.

Table 1. Main prototype hardware and functional roles.

Block	Typical element	Function
Meter interface	Electronic energy meter, 1000 imp/kWh	Produces pulses proportional to accumulated energy.
Isolation	PC817/opto-isolator + pull-up	Protects the controller and conditions the meter pulse.
Controller	ATmega328P-class MCU	Counts pulses, calculates kWh/bill, handles timing and communication.
Display & storage	16×2 LCD + EEPROM	Shows local readings and retains cumulative data after power loss.
Communication	SIM800/SIM900 GSM modem	Sends scheduled/on-demand readings by SMS or packet data.
Power/control	5 V regulator; optional relay driver	Supplies electronics and allows future remote disconnection.

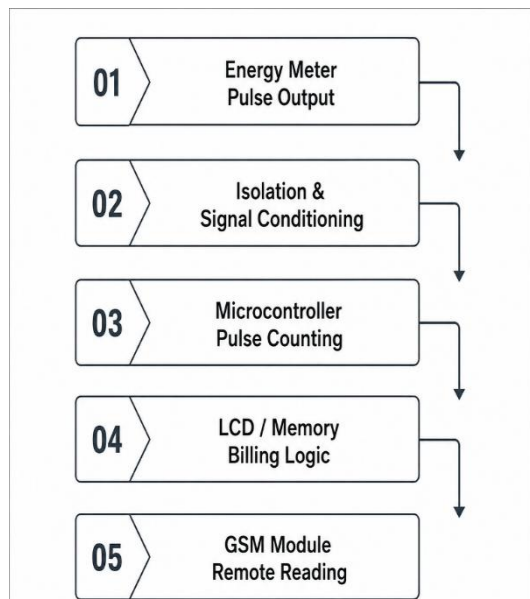


Figure 1. Functional architecture of the proposed microcontroller-based AMR system.

3. Pulse Acquisition and Energy Calculation

The conditioned pulse line is connected to an interrupt-capable input. Each valid rising edge increments the pulse count. Interrupt-driven acquisition is preferred to slow polling because it reduces the chance of missing narrow pulses while the program is updating the display or communicating with the modem. A short digital debounce or minimum inter-pulse time can reject spurious transitions. If N is the cumulative pulse

count and K is the meter constant in impulses per kilowatt-hour, accumulated energy E is computed as $E = N/K$ kWh. For $K = 1000$ imp/kWh, the resolution is 0.001 kWh per pulse. Billing can be represented by a simple tariff model $B = E \times R$, where R is the price per kWh. A commercial system may use block tariffs or time-of-use rules; however, a fixed tariff is sufficient for methodological verification. The latest cumulative count is written to EEPROM after a defined pulse increment or time interval instead of after every pulse, thereby reducing unnecessary write cycles. On restart, the stored count is recovered before new pulses are accumulated.

4. Communication and Firmware Sequence

The GSM routine is initialized after the metering task so that a temporary communication problem cannot stop local energy measurement. The controller checks modem readiness, network registration, and signal availability. At the reporting interval, it prepares a compact message containing meter ID, cumulative kWh, estimated bill, and timestamp. If transmission fails, the record remains stored locally and a retry counter schedules another attempt. This buffered approach follows the communication-reliability concerns identified in AMR research [7]. On-demand operation can also be implemented: an authenticated request containing a predefined command causes the controller to return the current reading. Ashna and George [9] demonstrated the value of combining GSM meter

access with centralized billing, while the 2017 design of Labib et al. [10] showed that authenticated cellular control can support richer smart-meter functions.

The firmware sequence is: initialize I/O, LCD, EEPROM, UART, and interrupts; restore the last pulse count; enter the main loop; update energy and billing values; display current status; check reporting timer; communicate when required; save state periodically; and continue pulse acquisition at all times. Optional extensions include tamper input monitoring, relay control, prepaid credit logic, power-failure notification, and remote tariff update.

5. Validation Procedure

Because this methodology paper does not claim a completed certified laboratory build, validation is performed using a controlled numerical pulse model rather than fabricated physical measurements. Six load-duration combinations are converted into reference energy using $E_{ref} = P \times t / 1000$, where P is watts and t is hours. The expected pulse count is $K \times E_{ref}$ and the

simulated meter output is quantized to the nearest whole pulse, reflecting the finite pulse resolution of a 1000 imp/kWh meter. The microcontroller reading is then $E_{mcu} = N_{pulse}/K$. Percentage error is $|E_{mcu} - E_{ref}|/E_{ref} \times 100$. This procedure isolates the counting and quantization behaviour of the proposed digital method and is reproducible from the stated equations.

Results

Table 2 reports the numerical validation. Cases whose theoretical pulse count is an integer produce no quantization error. Non-integer cases are rounded to the nearest whole pulse, producing small deviations that decrease as the accumulated energy increases. The largest observed deviation is 0.44% for the 75 W, 45-minute case, while the 750 W, 15-minute case has an error of 0.27%. Across all six cases the mean absolute percentage error is approximately 0.12%, corresponding to an average accuracy of about 99.88% for the pulse-counting stage under the stated assumptions.

Table 2. Numerical pulse-counting validation for $K = 1000$ imp/kWh.

Test	Load (W)	Duration (min)	Reference (kWh)	Pulses	MCU reading (kWh)	Error (%)
1	60	60	0.06000	60	0.06000	0.000
2	75	45	0.05625	56	0.05600	0.444
3	150	30	0.07500	75	0.07500	0.000
4	300	25	0.12500	125	0.12500	0.000
5	750	15	0.18750	188	0.18800	0.267
6	1200	12	0.24000	240	0.24000	0.000

The numerical result should be interpreted as the resolution limit of the digital pulse interface, not as a complete meter-accuracy certification. A real hardware prototype would add errors from the certified meter itself, optocoupler pulse shape, clock timing, noise, power interruptions, and missed transitions. The methodology reduces these risks through interrupt-driven acquisition, electrical isolation, minimum pulse-width checks, EEPROM backup, and separation of the metering task from the GSM task. The proposed architecture also preserves a local reading even when the network is unavailable, so communication failure need not imply loss of energy data.

Conclusion and Discussion

This paper presents a reproducible methodology for designing a microcontroller-based automatic energy meter reading system using the pulse output of an electronic electricity meter. The design separates metrology from data processing: the existing meter supplies energy-proportional pulses, while the microcontroller

provides counting, energy conversion, tariff calculation, local display, storage, and GSM communication. This approach is simpler than building an entire metering analog front end and is therefore appropriate for low-cost academic prototypes or retrofitting studies. The numerical validation shows that with a 1000 imp/kWh pulse constant the quantization contribution can remain below 0.5% for the tested cases, with a mean absolute percentage error of about 0.12%. These values do not replace laboratory certification but demonstrate that the pulse-resolution mechanism itself can support accurate automated reading. The literature up to 2017 shows that AMR evolved from short-range wireless acquisition [1], embedded billing and load control [2], and ARM/GPRS remote management [3] into mesh-network [4], wireless digital [5], GSM instant-billing [6,9], protocol-aware [7], web-connected [8], and multifunction smart-meter architectures [10]. The proposed methodology intentionally combines only the core functions needed for a reliable AMR prototype. Its main limitations are dependence

on the pulse interface of an existing meter, reliance on cellular network availability for remote delivery, absence of cryptographic message authentication in the basic version, and lack of physical EMC/accuracy testing. Future work should implement the hardware prototype, compare it against a calibrated reference meter over resistive and inductive loads, add secure authentication and data encryption, evaluate long-term GSM delivery reliability, and extend the firmware to tamper detection, prepaid metering, and demand-side load control.

References

- Koay, B. S., Cheah, S. S., Sng, Y. H., Chong, P. H. J., Shum, P., Tong, Y. C., Wang, X. Y., Zuo, Y. X., & Kuek, H. W. (2003). Design and implementation of Bluetooth energy meter. *Proceedings of the Fourth International Conference on Information, Communications and Signal Processing*, Vol. 3, 1474–1477.
<https://doi.org/10.1109/ICICS.2003.1292711>
- Maitra, S. (2008). Embedded energy meter—A new concept to measure the energy consumed by a consumer and to pay the bill. *Joint International Conference on Power System Technology and IEEE Power India Conference*.
<https://doi.org/10.1109/ICPST.2008.4745360>
- Li, K., Liu, J., Yue, C., & Zhang, M. (2008). Remote power management and meter-reading system based on ARM microprocessor. *2008 Conference on Precision Electromagnetic Measurements Digest*, 216–217.
<https://doi.org/10.1109/CPEM.2008.4574730>
- Cao, L., Tian, J., & Liu, Y. (2008). Remote wireless automatic meter reading system based on wireless mesh networks and embedded technology. *Fifth IEEE International Symposium on Embedded Computing*, 192–197.
<https://doi.org/10.1109/SEC.2008.57>
- Bharath, P., Ananth, N., Vijetha, S., & Prakash, K. V. J. (2008). Wireless automated digital energy meter. *IEEE International Conference on Sustainable Energy Technologies*, 564–567.
<https://doi.org/10.1109/ICSET.2008.4747071>
- Sehgal, V. K., Panda, N., Handa, N. R., Naval, S., & Goel, V. (2010). Electronic energy meter with instant billing. *2010 Fourth UKSim European Symposium on Computer Modeling and Simulation*, 27–31.
<https://doi.org/10.1109/EMS.2010.18>
- Khalifa, T., Naik, K., & Nayak, A. (2011). A survey of communication protocols for automatic meter reading applications. *IEEE Communications Surveys & Tutorials*, 13(2), 168–182.
<https://doi.org/10.1109/SURV.2011.041110.00058>
- Landi, C., Merola, P., & Ianniello, G. (2011). ARM-based energy management system using smart meter and Web server. *2011 IEEE International Instrumentation and Measurement Technology Conference*, 1–5.
<https://doi.org/10.1109/IMTC.2011.5944309>
- Ashna, K., & George, S. N. (2013). GSM based automatic energy meter reading system with instant billing. *2013 International Multi-Conference on Automation, Computing, Communication, Control and Compressed Sensing*, 65–72.
<https://doi.org/10.1109/IMAC4S.2013.6526385>
- Labib, L., Billah, M., Rana, G. M. S. M., Sadat, M. N., Kibria, M. G., & Islam, M. R. (2017). Design and implementation of low-cost universal smart energy meter with demand side load management. *IET Generation, Transmission & Distribution*, 11(16), 3938–3945.
<https://doi.org/10.1049/iet-gtd.2016.1852>