

Efficient power management unit for IIOT edge computing applications

Mukul Sharma Mukul (✉ marvelmukulsharma@gmail.com)

Satyug Darshan Institute of Engineering and Technology

Shiv Kumar Shiv

Satyug Darshan Institute of Engineering and Technology

Rashmi Chawla Rashmi

J.C. Bose University of Science & Technology, YMCA

Research Article

Keywords: Internet of things, Data Sharing, Power task, Algorithms

Posted Date: August 3rd, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-763330/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Abstract

Applications of Industrial Internet of Things (IIoT) are comprehended in mass domains, and this reflects the prominence of IoT. However, the main con of IoT applications is that it requires continuous power to be drawn by various components to operate all the time. Energy harvesting provides a potential solution to this problem in many applications; however, this continual energy usage remains main IoT con. The energy generated by ambient sources is inadequate to run system application, and a need for better Power Management Unit (PMU) persists. This research paper proposes a PMU framework for energy efficiency in edge computing application with three layers of IoT architecture. Design-build on this deployment, constitutes sleep and awake scheduling, with calculative idle intervals. The transitions of states exercise an energy-efficient solution for the entire system. The design is tested on ZYBO Zynq-7000 (XC7Z010-1CLG400C) ARM/FPGA SoC board for development and evaluation. Simulation results demonstrate the task scheduling of our proposed architecture and energy consumption details. In brief, the proposed design proves to be a beneficial option, in particular, surpassing kin designs for industrial automation.

Introduction

In today's scenario, it is not sufficient for the industry to make data-driven decisions; data empowerment is needed to necessitate business automation. The Industrial Internet of Things (IIoT) integrates networked sensors, software analytics, and machine learning with machine to machine (M2M) communication and helps to reshape entire manufacturing industries. This technology is useful in predictive maintenance of the industrial process. However, scaling of IIoT systems results in a high degree of complications, accounting for significant challenges, and viz. increase in energy consumption. As the number of connected devices rises exponentially, providing power in the different application set up will be a significant challenge, and a feasible, scalable option as efficient power management circuit and the low voltage operation of the entire design will be requisite. Latterly, multiple research work on challenges in IIoT systems has been reported [28]. Wu Weiwei et al. [2] proposed a technique to enhance utilization efficiency for big data in case of multiple tasking. Daniel Minoli et al. [3] reviewed the issue and strategies to improve Power over Ethernet (PoE) for in-building connectivity. Wang and his team [4] proposed three layers of hierarchical framework balance the output load. The shift state mechanism in the research works helps in the reduction of energy consumption. The research on energy efficient IoT was investigated by C. Estevez and team [5]. Huaiying Sun et al. formulated energy and time efficient task offloading and resource allocation [28]. S.F.Abedin and team adopted the architecture, compromising a cloud server, sense entities domain, RESTful service hosted networks, and user applications for energy-efficient IIoT [6]. J.Leu et al. [7] deployed a conceptual system framework to overcome the stumbling blocks of flexibility and extendibility. The heterogeneity and interoperability in the IoT architecture were addressed by C.Sarkar and team [8].

The high gain momentum in IoT resulted in the development of cyber-physical systems with enhanced process control, simulation capabilities, and higher-level modelling for smart cities [9-14]. The emergence

of remote sensing and wireless sensor nodes monitoring [27] has also paved way to large IIoT applications [15-17] viz. Auma electric valve actuator, actuators for heating/cooling, corrosion protection, meter applications. Most of the IIoT actuation system depends decisively on the driver nodes and smart devices. Hence, optimization of time-based prediction model [18-20] will result in reducing energy consumption in processing [28] and communications for IoT devices. IIoT applications start with conventional industrial devices such as steam traps, pressure relief valves, and heat exchangers. The applications take data from industrial sensors and add networking mechanisms and dashboards. The cue difference between consumer and industrial capabilities in this context is the need for open standards and much higher expectations of reliability and security.

When applied to real-time Industrial Internet data streams, edge or cloud-based analytics can detect anomalies, essential events of triage, direct, prescriptive controls, signal predictive maintenance alerts, and more. Past research works have reported the use of ambient resources to harvest energy for IoT application as regular energy usage. With a lot of industry buzz around applied analytics and machine learning, it can be challenging to strategize a roadmap of powerful analytic applications to drive specific business outcomes. The goal of this paper is to reduce power consumption problem where a sensor node needs power for complete duration of time even if they are not sample (and transmit) the data. Note that the power consume in time T is processing power i.e. static, dynamic, logic switching, and capacity of the task. The proposed power management unit validates efficient reduction in the overhead power consumption. Further on this prediction model correlation between latency and throughput rate is worked at different standard clock frequencies.

In this paper, we depict the optimal power management circuit design for IIoT operation for industrial automation. The three-layer architecture for node deployment and actuation system is adopted, as shown in Fig. 1 with three subsets (i) Control layer (ii) Gateway layer and (iii) Perception layer comprising (Sensor Nodes) SN, (Gateway Node) GN, (Control Node) CN [19-21]. SN constitutes a group of sensors data providers that can be designated as a central platform to perform IoT operations. The business intelligence on data analysis is applied for energy management decisions for utility alike applications in industry [22, 23]. The analysed data is than transmit to control layer to prepare a predictive model so that actionable information in terms of working mode (subdivided into sleep mode and awake mode) for further used in actuation system [26]. The time based calculation on the power management unit along with the analytics engine can work on this data and prepare a flexible on-demand supply prediction model for the actuation system. The time-based throughput rate and latency analysis can further help in adjusting and optimizing power consumption. Also to this seamless remote access through IoT will help to control the edge computing application and get access to ON/OFF of the device via mobile phone/laptop/personal computers.

The contributions of our paper are summarized as follows:

- A system model for energy-efficient IoT is adopted, which consists of a sense entities domain using Message Queuing Telemetry Transport (MQTT) protocol in gateway layer of architecture.

- In the sense entities domain, we design a three-layer hierarchical framework to realize the deployment of nodes in IoT, achieving the goal of saving energy and increasing network lifetimes.
- Based on the aforementioned architecture, we develop an activity schedule based on Power Task Clustering (PTC) algorithm to switch nodes to sleep mode and wake them up when required based on the calculation of sleep interval.
- The latency calculations are further done on various frequencies and prediction model for the same build.

The remainder of our article is organized as follows. The proposed system model developed for energy efficient IIoT discussed in the section 2 consisting of the three-layer framework for the deployment of nodes in the proposed model and demonstrate the activity scheduling and wake up mechanism based on the predicted sleep intervals. Section 3 describe the implementation of the design including simulation results succeeded by timing and power report in section 4. Finally, section 5 provides conclusion and future work.

Proposed Pmu For Energy Efficient IiOT Architecture

The workflow model to in-built proposed PMU in Industrial IOT scenario is illustrated in cumulative description in Fig.2. The section of perception layer deals with data analysis and faces lot of challenges. For example, if we consider the industrial data of power consumption in a plant or industry with respect to the product portfolio [24,25] following points are requisite:

1. Precise volume data of energy consumption.
2. An accurate and precise record of historical production data.

The information of the above two data sources is reconciled before applying any machine learning algorithm as a business intelligence tool. The energy measurement units comprising sensors are needed to be installed to acquire this collective data chain for the first point. This will help in building the Power Task Clustering (PTC) algorithm. The second point faces myriad challenges as real-time data analysis is required. The manual data entry is added with inaccuracies and starts and end time estimation needs proper framing to figure out sleep and awake intervals. In this work following perturbation viz. random fluctuation in sensor data, reduced sensor data frequency and imprecise manufacturing execution system are not considered.

The acquired industrial data can now be analysed by using business intelligence via PTC algorithm to check on power consumption and sleep/awake time interval can figure out. These figured data intervals are the critical requirement for the designed PMU in this architecture. The PTC algorithm (flowchart for same illustrated in Fig. 2(b)) employed to schedule task for PMU is enlisted below:

1. Initially, all data are marked unexamined, and each task forms a separate cluster.
2. Sort all task in a decreasing order according to their duration time.

3. Repeatedly merges task by zeroing the data with high duration time if total power consumption is not increased.
4. Mark the data examined.
5. When two clusters are merged, the task is ordered accordingly to their latency and throughput.
6. Until all data are marked examined.

To figure out sleep and awake time intervals task scheduling is discussed in next section.

2.1 Task Scheduling

The IIOT perception layer data request is modelled as a task in which the time is partitioned into discrete time slots, where s_i is start time, e_i is end time. w_i is a time slot for workload ($e_i - s_i$). time require by a task can be define by function of (s_i, e_i, w_i) . Without loss of generality, we assume that $\min s_i = 0$ and $\max e_i = T$. for each task in time duration of $(T_{s_i}, T_{e_i}]$ power is allocated as $P_{a_i}, l = 1, 2, \dots, n$. which is the power required in awake mode and P_{s_i} power required in sleep mode.

Parameters considered for task scheduling are:

T_{s_i} - start time

T_{e_i} – end time

T_{f_i} – time duration ($e_i - s_i$)

P_a - power in awake mode (for T_{f_i} time, i.e., $P_{a_{e_i}} - P_{a_{s_i}}$)

P_s - power in sleep mode (for T_{f_i} time, i.e., $P_{s_{e_i}} - P_{s_{s_i}}$)

l - power for idle scheduling (i.e., power required in checking for the arrival of a new task) function for $s_i, e_i,$ and f_i .

- In sleep mode task is not performed, but static, dynamic and clock switching power $P_s (s_i, e_i, f_i)$ is require in gateway layer.
- At time $T_{a_{s_i}}$ a task is executed in awake mode, in perception layer and power is consumed till $T_{a_{e_i}}$ (awake end time), so power consumed in awake in mode is $P_{a_i} (i=1,2, 3..)$.
- After the task was finished, checking/sensing is performed for the execution of another, and it assumed that power requires for sensing is $l (s_i, e_i, f_i)$.

For a given scheduled time duration for $T(s_i, e_i, f_i)$ power is summation of P_s for sleep mode; $P_{a_i}, (i=1,2,3..)$ for awake mode and l for idle schedule, as idle schedule is for short time ($T_{f_i} \approx 0$) it can be neglected.

The total power requirement (eq.1) for task scheduling is $\sum_{i=1}^n P_{ai} + \sum_{i=1}^n P_s$ as illustrated in Fig. 3 however $P_{ai} \gg P_s$.

$$P(T) = \sum_{i=1}^n P_{ai} \quad (\text{eq.1})$$

The primary intent of the prediction model for task scheduling is to optimize power by making actuation system work in two modes: sleep and awake. In the sleep mode nodes of actuation system are not working, they can be switched on to awake mode as and when required. The prediction model of sleep interval is considered and analysed for latency and throughput calculation for further optimization. The idle time for awake mode can be set as per the requirement, and the actuation system is transferred to sleep mode in case no work done during this time period. The objective is to switch some nodes to sleep mode when they are not in a working state and wake them up when required. However, the sleep interval of the nodes is impacted by the arrival of the new task.

The next section works on PMU (a subset of control layer) to align the task scheduling in terms of sleep and awake mode.

2.2 PMU Hardware Specifications

The hardware architecture of the proposed PMU constitutes various building blocks, as illustrated in Fig. 4 and are described below:

1. Actuation system to monitor and control the AC power units, an integrated switches/relays and optocouplers are interfaced with the microcontroller in context to the required application [20]. A solid-state relay used to turn on/off a device is controlled by a microcontroller.
2. High-end Microcontroller-is deployed for analysis and processing on data. A Zynq -7000 (AP SoC/FPGA) is used in the high end as an edge device data acquisition module to perform an experimental analysis of the proposed design. The specifications of the SoC are displayed in Table 1
3. Servers: are high-end PCs that can be further be exploited on Cloud for wide-scale comprehensibility. The installed servers are MQTT Broker, a Web Server, Analytics Engine server, and highly scalable Storage Server.
4. User Application-Working as client-side applications, the user applications can be classified into the following two categories according to the authentication mechanisms.

Client Applications: To use virtual objects hosted as applications in the virtual environment, the client uses the application interface to send requests to the server. Direct access is not allowed within this authentication.

Admin Applications: Unlike the client application, an admin application has the right to access the server directly. Then the admin can promptly make necessary modifications to the system and monitor the performance of the whole system.

Implementation Of Proposed Design

To validate the architecture of the proposed system, the PMU design was tested on ZYBO Zynq-7000 (XC7Z010-1CLG400C) ARM/FPGA SoC board for development and evaluation. The implemented design employs a counter to count the time interval for sleep and awake mode. The system will go to sleep state if no response is given to the design for a specific time duration. At the awake mode different jobs (represented as Task 1, Task 2 and Task 3 as shown in Fig. 5) are executed, in perception layer of the architecture and after that, the state will go to an idle mode then sleep mode. Fig. 5 illustrates the simulation waveform to represent a different state of PMU and task executed; after completion of each task, PMU goes in idle state. The idle state analysis two conditions for state transition (i)perform the Task occur during idle state (ii)if no Task happen then go to sleep mode.

Considering industrial application, the variation in awake time interval are enlisted in Table 2. The difference in output parameters: Latency and Throughput rate will help in figuring the optimized time interval of the task. As the proposed unit is a model for IIOT applications small time intervals are considered for ease of computation and Worst Hold Slack (WHS) and Worst Negative Slack (WNS) for the design is also enlisted. The erstwhile detail worst hold/removal slack and later provides the worst setup/recovery slack information.

Result Analysis

Timing and Power report:

Timing report shown in Table-3 exhibits total time required for set up, hold, and pulse width. The report provides high-level information on the timing characteristics of the design compared to the constraints provided viz. Total Negative Slack (TNS) which is the sum of the setup/recovery violation for each endpoint in the entire design or a particular clock domain and WNS. Total Hold Slack (THS) is the sum of the hold/removal violation for each endpoint in the entire design or a particular clock domain. Power report illustrated in Table-4 shows the total power of designed and on-chip static and dynamic power required and also represent a different amount of power needed by a clock, logic, and input changes. It represents the power consumed by a task in the mode (sleep and awake). When a power management unit is in a sleep mode, then power consumed is determined by static power (Total Power – Dynamic Power). If the task is executed in awake mode, then consume power is determined by dynamic power.

The default setting of the simulation is as follows, tasks start time in awake mode is considered by the power aware task clustering, for each start time T_{si} , the end time T_{ei} is generated. All workload time T is sorted so that an earlier arrived task carries an earlier end time, which is $T_{ei} - T_{si}$. In our case the number of task is set to be 3. The maximum length of workload request is mention in Table 2 with WHS, WNS, Latency and throughput. We first evaluate the performance when the number of tasks increases from 10 to 20 min with a step of 10 and so on upto 200 min. the total power consumption of our design is remain consistent to our analysis as the module goes to sleep mode whenever task duration is complete.

This report presents the Dynamic power and activity for each resource used in the PMU. Each resource is sorted by the amount of power; the highest power consuming nodes will be listed first. Definitions:

Clocks – shows power for each clock tree in the design.

Signals – shows activity rate and power for the design routing elements.

Logic – shows activity rate and power for the design slice element (for example, LUTs, flip-flop).

IO – shows activity rate and power for the different input outputs in the design.

The design is optimized by reducing the number of counters which bring down the total power consumption. When the power management unit is in sleep mode power consumes remain static power but when a task is performed in awake mode dynamic power reduce because by using single counter power consumed by logic, net, signal is less. This is illustrated in Table-4. Further by optimization, we reduced the number of a flipflop, LUT required (only 41 Flip-flops, 25 LUTs and 2.477 WNS is observed).

We have further examined the effect of frequency on the latency and throughput (subsets of gateway layer) which is illustrated in Table-3. Next the correlation between latency with throughput for different awake duration (code illustrated in Fig. 6) is expressed as eq (2). The correlation setup will help in throughput justification at different frequencies for practical implementation of the proposed research work. Here, we have not compared the transmission delay by increasing the workload time duration as no optional solution is known in the literature for same. Therefore, the simulation results, power and time report validate the efficiency and reduction in power consumption in three-layer architecture of our PMU design.

$$\text{Latency} = 12270 * \text{throughput} - .002436 * \text{Frequency} + 2.164 \quad \text{eq (2)}$$

Conclusion

This software-centric implementation reported works on a predictive Platform that provides a unique framework to control Power consumption. The main intent of the prediction model is to optimized power by making actuation system work in two modes: sleep and awake. The Power Task Clustering (PTC) algorithm is built to switch nodes to sleep mode and wake them up when required based on the calculation of sleep interval. But it also offers a unique framework to create machine learning analytics. These analytics assist organizations in predicting critical failures, automating the process, and driving efficiencies. The correlation analysis can be further deployed for selecting controller clock frequency and latency. From the perspective of energy saving in industry, we adopted an architecture composed of a sense entities domain; service hosted networks, a cloud server, and user applications. Then, to the massive deployments of nodes in the sense entities domain, we presented a three-layer architecture that includes a sensing layer, a gateway layer, and a control layer. By forbidding direct communication between two SNs and using GNs as relay nodes, the proposed structure may save energy and prolong the lifetime of the system. Moreover, regarding the nodes, a sleep scheduling and wake up protocol has been

submitted. By calculating the sleep interval of SNs, the GN can change the state of SNs for efficient energy utilization. Meanwhile, the CN decides the allocation of SNs to GNs. An evaluation has validated the effectiveness of our architecture in improving resource utilization and energy consumption.

Declarations

Conflict of Interest

On behalf of all authors, the corresponding author states that there is no conflict of interest.

Data Availability Statement

The authors declare that the data supporting the findings of this study are available within the article.

References

1. W. Wu, X. Zhai, and Y. Zhao, "On Minimizing Sensing Time via Data Sharing in Collaborative Internet of Things," in *IEEE Access*, vol. 6, pp. 41633-41642, 2018.
2. Daniel Minoli et al., "IoT Considerations, Requirements, and Architectures for Smart Buildings Energy Optimization and Next Generation Building Management Systems," in *IEEE Internet of Things Journal*, Vol.4, no.1, pp. 269-283, Feb. 2017.
3. K. Wang, Y. Wang, Y. Sun, S. Guo and J. Wu, "Green Industrial Internet of Things Architecture: An Energy-Efficient Perspective," in *IEEE Communications Magazine*, vol. 54, no. 12, pp. 48-54, Dec. 2016.
4. C. Estevez and J. Wu, "Recent Advances in Green Internet of Things," in Proc. *IEEE Latin-American Conf. Commun. (LATINCOM)*, pp. 1–5, Nov. 2015.
5. S. F. Abedin et al., "A System Model for Energy Efficient Green-IoT Network," in Proc. *Int'l. Conf. Information Networking (ICOIN)*, pp. 177–82, Jan. 2015.
6. C. Sarkar et al., "A Scalable Distributed Architecture Towards Unifying IoT Applications," in Proc. *IEEE World Forum on the Internet of Things (WF-IoT)*, pp. 508–13, Mar. 2014.
7. J. Leu, C. Chen, and K. Hsu, "Improving Heterogeneous SOA-based IoT Message Stability by Shortest Processing Time Scheduling," in *IEEE Trans. Services Comp.*, vol. 7, no. 4, pp. 575-85, Oct. 2014.
8. Luigi Atzori, Antonio Iera, and Giacomo Morabito, "The internet of things: A survey Computer networks," vol. 54, no. 15, pp. 2787–2805, 2010.
9. Chawla, R., Singal, P., and Garg, A. K., "A Mamdani Fuzzy Logic System to Enhance Solar Cell MicroCracks Image Processing," *3D Res.*, vol.9, no.34, pp.1–12, 2018.
10. Amir H. Alavi, Pengcheng Jiao, William G. Buttler, Nizar Lajnef, "Internet of Things-enabled smart cities: State-of-the-art and future trends," in *J. of Measurement*, vol. 129, pp. 589-606, 2018.
11. Yuan fang Chen, Gyu Myoung Lee, Lei Shu, and Noel Crespi, "Industrial internet of things-based collaborative sensing intelligence: framework and research challenges," in *Sensors*, vol. 16, no. 2,

2016.

12. Makkar, A., Raheja, A., Chawla, R., and Gupta, S., "IoT Based Framework: Mathematical Modelling and Analysis of Dust Impact on Solar Panels," *3D Res.*, vol.10, no.3, pp. 1–15, 2019.
13. Xiaolin Fang, Hong Gao, Jianzhong Li, and Yingshu Li, "Application aware data collection in wireless sensor networks," in *INFOCOM*, Proc. IEEE, pp. 1645–1653, 2013.
14. Chawla R, Singhal P, Garg A. "Internet of Things Driven Framework for Smart Solar Energy System" *ASME. J. Energy Resour. Technol.* vol.142, no.1, pp:011201-011201-9, 2019.
15. Zhenni Feng, Yanmin Zhu, Qian Zhang, Lionel M Ni, and Athanasios V Vasilakos, "Trac: Truthful auction for location-aware collaborative sensing in mobile crowdsourcing," in *INFOCOM*, Proc. IEEE, pp. 1231–1239, 2014.
16. Raghu K Ganti, Fan Ye, and Hui Lei, "Mobile crowdsensing: current state and future challenges," in *IEEE Communications Magazine*, vol. 49, no. 11, 2011.
17. William I Grosky, Aman Kansal, Suman Nath, Jie Liu, and Feng Zhao, "Senseweb: An infrastructure for shared sensing," in *IEEE multimedia*, vol. 14, no. 4, 2007.
18. Weiwei Wu, Jianping Wang, Minming Li, Kai Liu, and Junzhou Luo, "Energy-efficient transmission with data sharing," in *Computer Communications (INFOCOM), IEEE Conference*, pp. 73–81, 2015.
19. Murtaza A Zafer and Eytan Modiano. "A calculus approach to energy efficient data transmission with quality-of-service constraints," in *IEEE/ACM Transactions on Networking (TON)*, vol. 17, no. 3, pp. 898–911, 2009.
20. Andrea Zanella, Nicola Bui, Angelo Castellani, Lorenzo Vangelista, and Michele Zorzi. "Internet of things for smart cities," in *IEEE Internet of Things journal*, vol. 1, no. 1, pp. 22–32, 2014.
21. Weiwei Wu, Xiangping (Bryce) Zhai, Yingchao Zhao, "On Minimizing Sensing Time via Data Sharing In Collaborative Internet Of Things," in *IEEE Access*, vol. 6, 2018.
22. Yawei Zhao, Deke Guo, Jia Xu, Pin Lv, Tao Chen, and Jianping Yin, "Cats: Cooperative allocation of tasks and scheduling of sampling intervals for maximizing data sharing in WSNS," in *ACM Transactions on Sensor Networks (TOSN)*, vol. 2, no. 4, 2016.
23. Chun sheng Zhu, Victor CM Leung, Laurence T Yang, and Lei Shu, "Collaborative location-based sleep scheduling for wireless sensor networks integrated with mobile cloud computing," in *IEEE Transactions on Computers (TOC)*, vol. 64, no. 7, pp. 1844–1856, 2015.
24. Chun sheng Zhu, Joel J.P.C.Rodrigues, Victor C.M.Leung, Lei Shu, and Laurence T. Yang, "Trustbased communication for the industrial internet of things," in *IEEE Communications Magazine*, vol. 56, no. 2, pp. 16–22, 2018.
25. Chun sheng Zhu, Huan Zhou, Victor C. M. Leung, Kun Wang, Yan Zhang, and Laurence T. Yang.
26. Seokhoon Kim, Dae-Young Kim, "Efficient data-forwarding method in delay-tolerant P2P networking for IoT services", in *Peer-to-Peer Network, Appl.* vol.11, no.6, pp. 1176-1185, 2018.
27. Illsoo Sohn, Sang Won Yoon, Sang Hyun Lee, "Distributed scheduling using belief propagation for internet-of-things (IoT) networks", in *Peer-to-Peer Network, Appl.* vol.11, no.1, pp. 152-161, 2018.

28. Huaiying Sun, Huiqun Yu, Guisheng Fan, Liqiong Chen "Energy and Time Efficient task offloading and resource allocation on the generic IoT-fog-cloud architecture", in *Peer-to-Peer Network, Appl.* pp. 1-16, 2019.

Tables

Due to technical limitations, Table 1,2,3.4 and 5 are only available as a download in the Supplemental Files section.

Figures

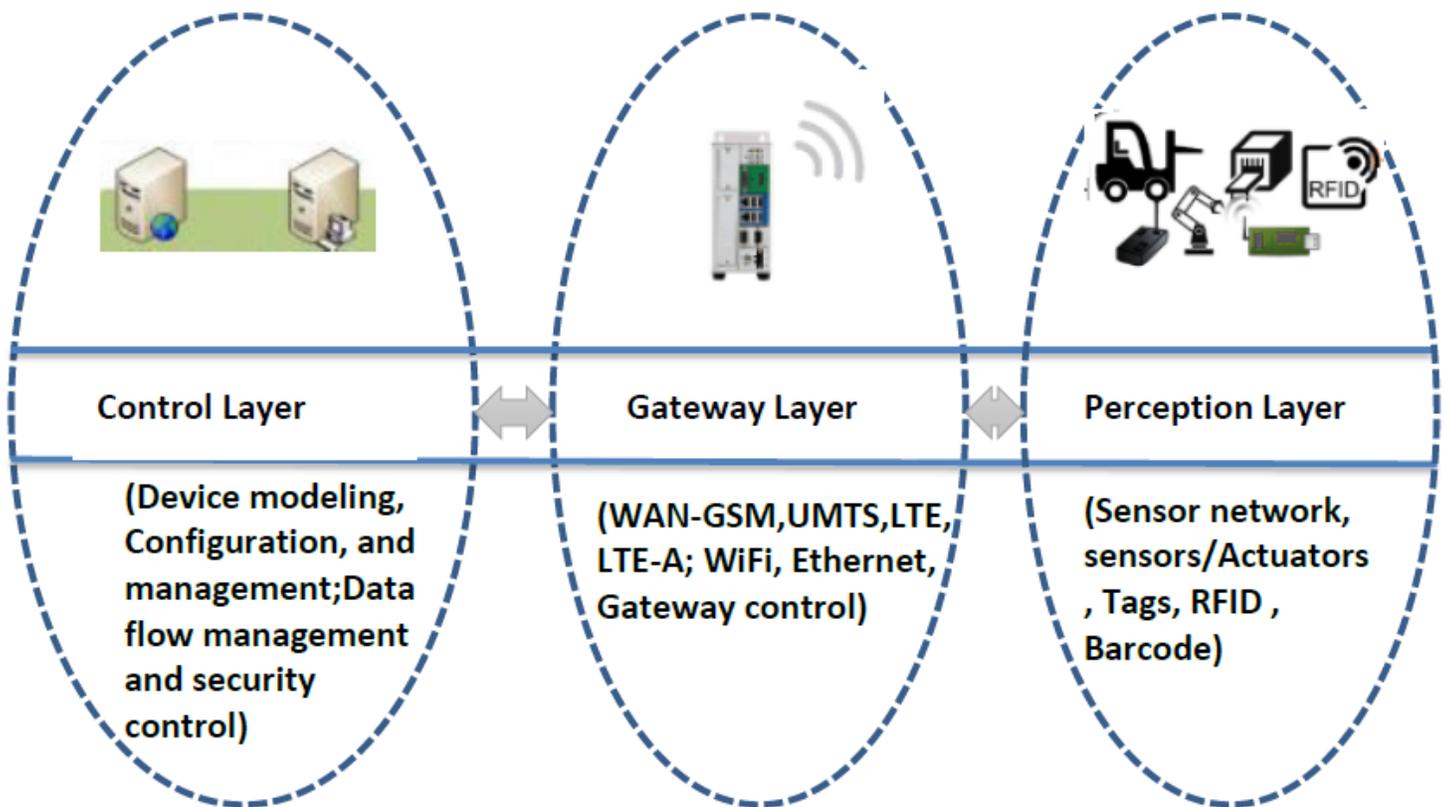


Fig.1: The IoT three-layer architecture of sensing entities domain in home automation.

Figure 1

See image above for figure legend.

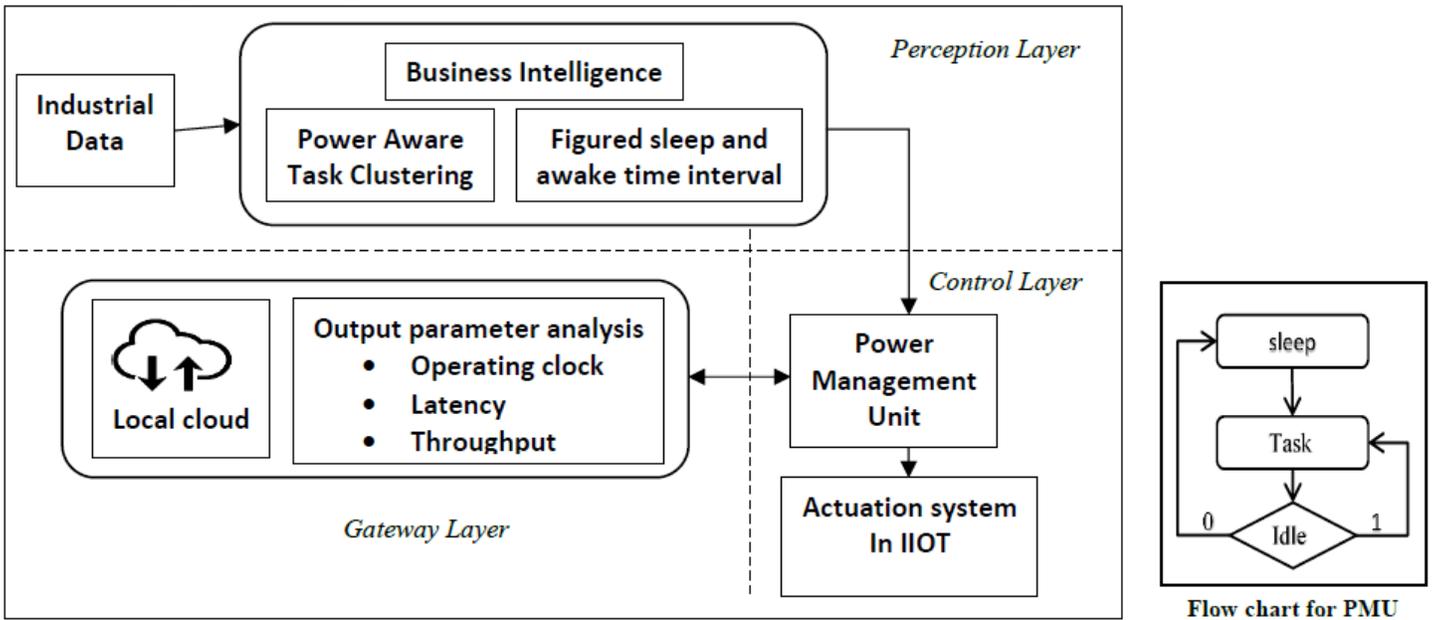


Fig. 2 PMU for energy efficiency IIOT Architecture

Figure 2

See image above for figure legend.

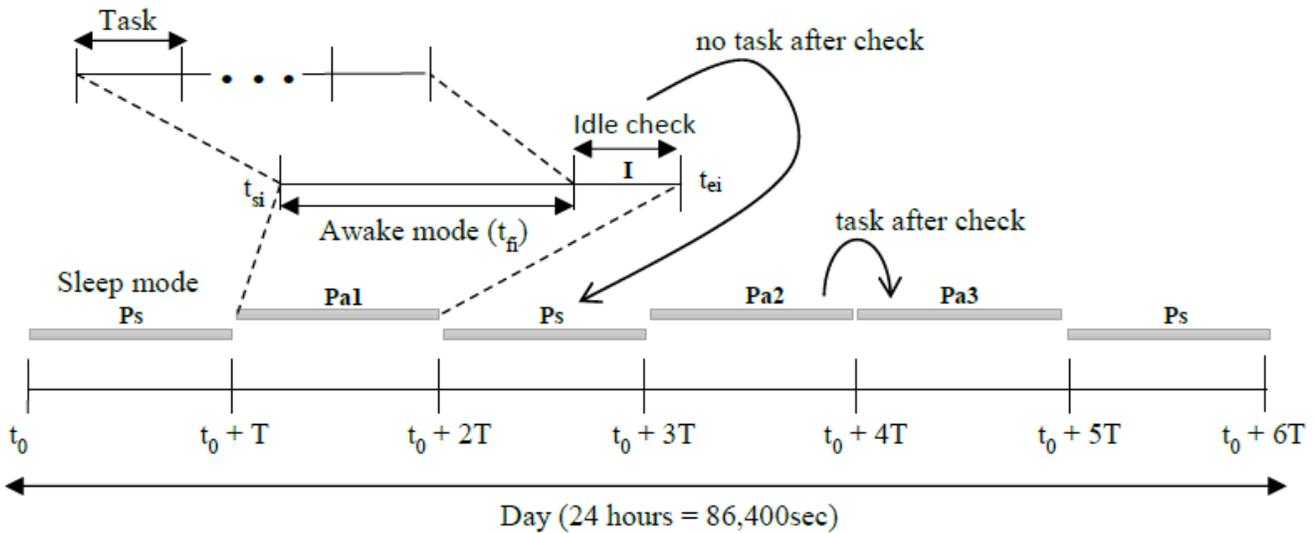


Fig. 3: Power Task Clustering adopted in proposed PMU

Figure 3

See image above for figure legend.

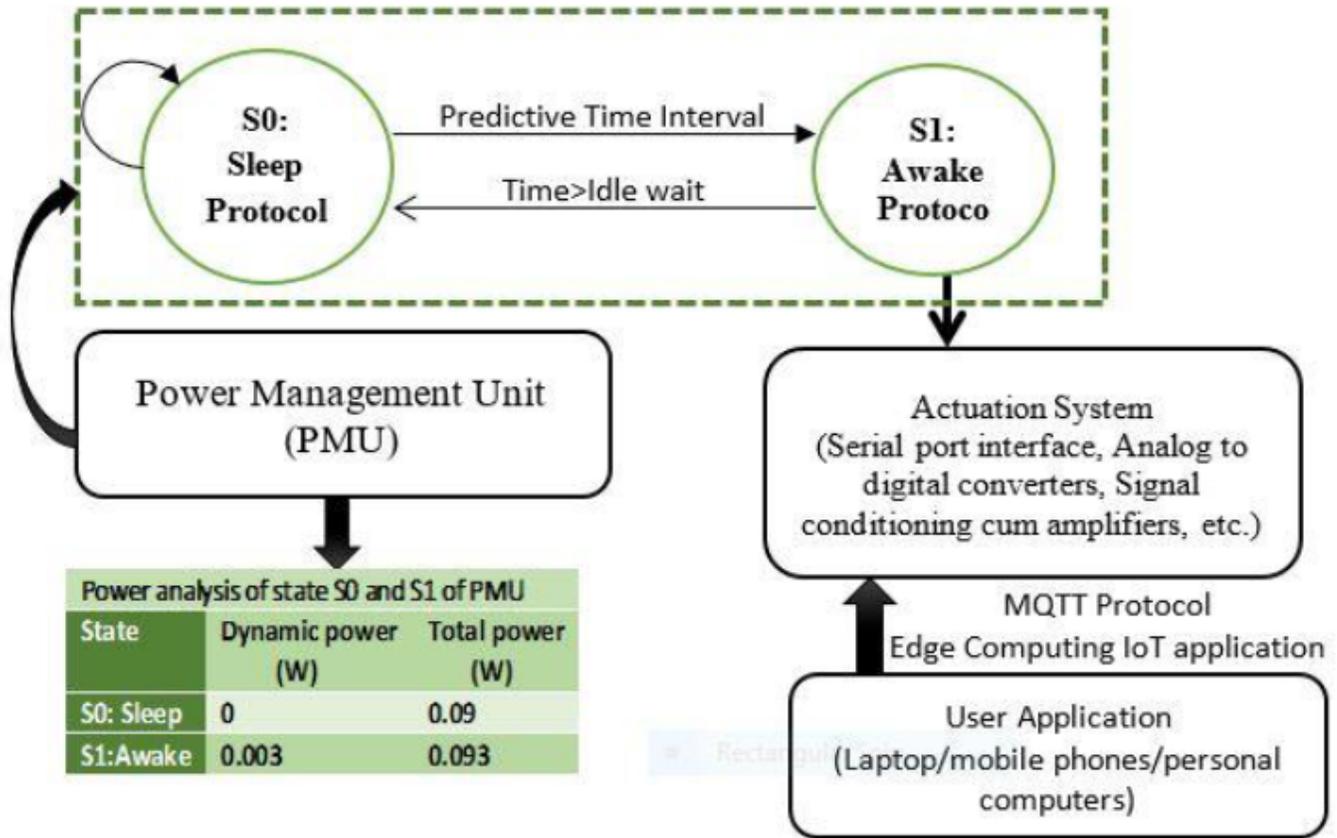


Fig. 4: The proposed IIoT energy efficient model

Figure 4

See image above for figure legend.

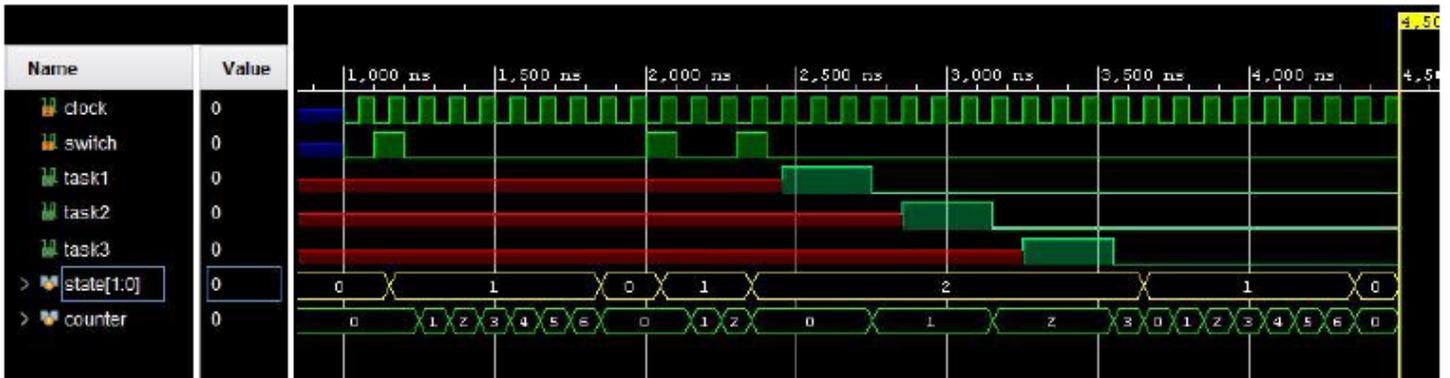


Fig. 5: Simulation result of proposed PMU

Figure 5

See image above for figure legend.

```
> t <- lm(data = final_ymca, Latency~Throughput+Frequency)
> t

Call:
lm(formula = Latency ~ Throughput + Frequency, data = final_ymca)

Coefficients:
(Intercept)  Throughput  Frequency
  2.164e+00   1.227e+04  -2.436e-02
```

Fig 6 Correlation code

Figure 6

See image above for figure legend.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [table1.png](#)
- [table2.png](#)
- [table3.png](#)
- [table4.png](#)
- [table5.png](#)