

# Hardware Layer of Ambient Intelligence Environment Implementation

Ales Komarek, Jakub Pavlik, Lubos Mercl, and Vladimir Sobeslav<sup>(✉)</sup>

Faculty of Informatics and Management, University of Hradec Kralove,  
Rokitanskeho 62, Hradec Kralove, Czech Republic  
{ales.komarek,jakub.pavlik.7,lubos.mercl,vladimir.sobeslav}@uhk.cz

**Abstract.** Ambient Intelligence is growing phenomena caused by advances of speed and size of computational hardware. It is possible to collect data from various sensors, devices or services and react to evaluated event and start the predefined process. These systems are closely connected with Cloud based services as they provide publicly available interface for management and visualization, computational and storage capabilities for forecasts and other advanced data processing to create ambient intelligence environment. The true ambient environments react to the presence of people and the sensors and actuators have become part of the environment.

This article presents RoboPhery project aimed to provide abstraction layer for interfacing any low cost hardware sensors and actuators with MQTT and Time-series Database bindings, that can serve as sensory network for Ambient Intelligence environments. The service architecture is designed to be so simple at hardware level to support single-board micro-controllers like ESP2866, ESP32 modules as well as single-board computers based on ARM or x86 architectures. The communication among individual devices is handled by the standard MQTT messages. The same devices can be configured to support multiple use-cases on configuration level to lower the operational costs of the solution.

**Keywords:** Automation · Sensor · Actuator · Service-oriented architecture · Ambient Intelligence

## 1 Introduction

The RoboPhery project makes it possible to utilize inexpensive Linux and MicroPython compatible micro-controllers, hardware actuators and sensors to create fully autonomous Ambient Intelligence platform. This platform can perform wide range automation use cases. These range from simple environmental automations, surveillance, to complete ambient intelligence environments.

First chapter introduces global architecture of autonomous agents with communication protocols for messages to the overlaying event driven controllers, dashboards or any other services. These systems use scalable and modular architecture to minimize the computational overhead which allows efficient use of available hardware resources.

The next part covers the autonomous agents in more details. Models of hardware interfaces and corresponding modules. It also covers description of the virtual modules that provide the simple thresholds or fuzzy logic reasoning for autonomous control.

The final chapter shows simple interaction with overlay control system that can communicate with multiple autonomous systems, gather vital information and even publish arbitrary actions on demand.

The control system contains time-series database, dashboard and reasoning platform which can detect hardware malfunctions of autonomous agent systems and perform necessary steps to repair it. With the usage of virtual models that provide high-level access to individual physical modules of individual agents.

### 1.1 Ambient Intelligence Environments

The main and basic ideas of Ambient Intelligence is an interconnecting an environment and technologies (mainly interconnected sensors and devices). This whole system can be designed to take care of this environment based on real-time and historical information (data). Ambient Intelligence inherits aspects of many areas of Computer Science, see Fig. 1, but should not be confused with any of those technologies [1, 14, 15].

An example of ambient intelligence is a Home environment, where is possible to solve some a lot of challenges using ambient intelligence [9]. Ambient intelligence was originally pointed by Eli Zelkha at Palo Alto Ventures in the late 1990s. This was a future vision of consumer electronics and computing and time frame for this was 2010–2020 [1, 20].

In an ambient intelligence, devices are interconnected to people support in life activities everyday, tasks and rituals in a natural way that uses information from physical environment collected by network connected devices and uses cloud power to drive the intelligence that can react and learn to the information coming from the environment [1, 5, 19].

The physical devices are constantly smaller and integrated into environment and the technology is mainly visible for end users just only via user interfaces (dashboards) [10].

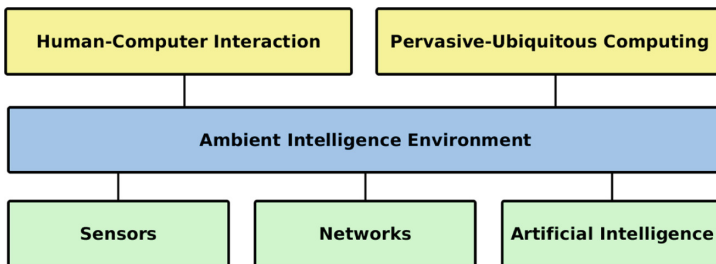


Fig. 1. Ambient Intelligence relationship to other computing science areas

The ambient intelligence is built on a pervasive and ubiquitous computing, profiling, context awareness, and design on human-centric computing. Also it is characterized by interconnected systems and technologies [1, 5, 15, 20] (Table 1).

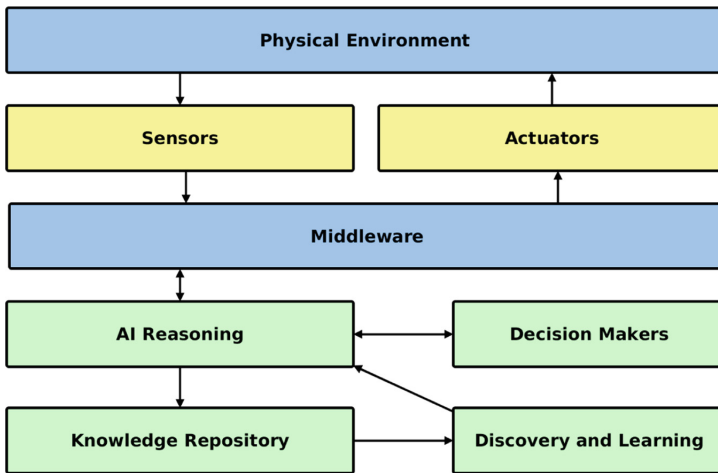
**Table 1.** Main Ambient Intelligence environment properties

Property	Description
Embedded	A lot of interconnected devices in the environment
Context aware	React to current context
Personalized	Meet the requirements of users
Adaptive	React to current state and context in the environment
Anticipatory	Anticipate user desires

### 1.2 Information Flows

There is a lost of was for building an Ambient Intelligence systems or environments, but the most often there are needed sensors and devices to surround users of an environment (interactors) with technology [5].

The technology collects and provides accurate data to the system. The collected data are transferred over the network and processed by middleware, which can collect data from multiple devices [4, 18] (Fig. 2).



**Fig. 2.** Flow of information in Ambient Intelligence system

An decision-making can be more beneficial to the people in the environment, when the system will provides higher level of reasoning and will accomplish diagnosis and advise, or system will assist humans with responsibility for intervention [4, 16].

High level Decision Making service are composed of a Knowledge Repository, where the events from environment are collected and stored. Thanks of this, AI Reasoner applies for example spatio-temporal reasoning to make decisions based on collected data.

Decision trigger actions to be performed in the environment and this is enabled via Actuators. The Knowledge discovery uses machine learning techniques to learn from the acquired data and information in order to update the AI Reasoner with the gained experience of the system. A typical information flow of Ambient Intelligence system is depicted in Fig. 3.

## 2 Ambient Intelligence Environment System

RoboPhery project provides software drivers for a wide variety of commonly used sensors and actuators through both wired and wireless interfaces. Platform is designed for multiple interfaces to be stacked (GPIO, IC or SPI over I2C or Bluetooth) to provide seamless integration with large variety of hardware platforms. The software drivers interact with the underlying hardware platform (or micro-controller), as well as with the attached sensors, through API calls to the interfaces [1, 3, 8, 17].

The actuators are controlled and the data is collected on RoboPhery Managers and this functionality forms a hardware foundation for ambient intelligence environment. The event-driven control system, data storage and data processing systems are provided in form of external cloud services or handled by computer with more processing power [7].

On the Fig. 3, you can see components of our proposed ambient intelligence system and their relationships. The central component is the message bus, in our case provided by MQTT broker [7, 11].

The basic architecture consists of time-series database, user dashboard and event-driven control engine, machine learning services and the on premise RoboPhery Manager. The actual RoboPhery service is written in Python in way compatible with MicroPython used on micro-controllers.

### 2.1 Global Message Bus

MQTT is a machine-to-machine connectivity protocol in area of “Internet of Things” [12]. MQTT was designed as an extremely simply publish/subscribe messaging transport protocol and thanks of this, it can be used for connections with remote locations where a small code footprint. This is reason, why requirements and/or network bandwidth are low. The MQTT broker can handle thousands of messages per second, supports high-availability setups for both high performance and stability. Individual autonomous agents and cloud-based Control system along with time-series databases are connected to this common message bus [11].

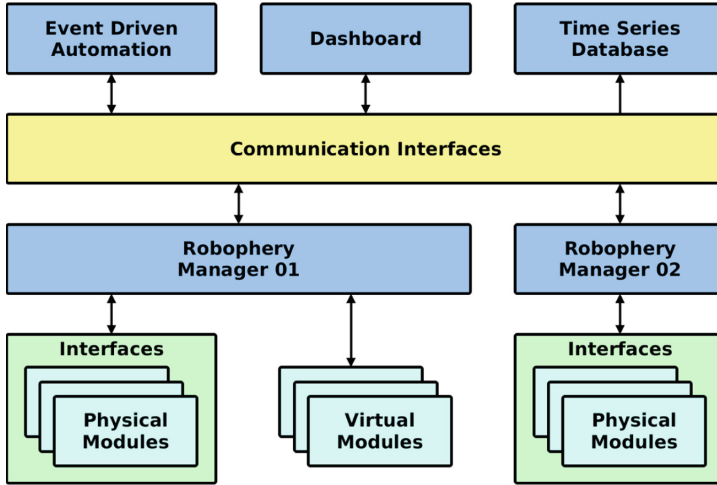


Fig. 3. Ambient Intelligence environment system

## 2.2 Cloud-Based Controller

As a architecture can be used an Event-driven architecture (EDA), which is message-driven architecture, where all components are reacting to events [13].

Events are significant changes in a state of components in the environment. For example, when a user turns on a computer, the state of computer changes from “off” to “on” and this change is captured as an event [7]. After this system architecture will treat this state change as an event which can affect other components in the environment. Also based on this change can be created message called the event notification. Events just occur, notification can be provide to the environment.

**Time-Series Database.** A time series databases are optimized for time series data handling and storage, where numbers are indexed by time (a date time or a date time range) [6,8]. A time series of storage prices is called a price curve and energy consumption is called a load profile.

Database queries and transactions are used and useful for analyzing, but these operations should be specialized for time-series data for correct, reliable and effective implementation of an time-series database. Time-series Databases are optimized for time series data.

Complex softwares for time series data may not be practical with regular relational database systems. Traditional flat file databases are not appropriate, whe the data and a count of transactions reaches a maximum threshold determined by the capacity of individual servers. Queries for historical data, with time ranges and time zone conversion are difficult in a traditional relational database.

The problem for traditional relational databases is that they are often not modeled correctly to respect time series data and time-series databases impose a model and this allows them to provide more features for time series data.

**Machine Learning Engine.** Machine learning is science discipline, which deals with algorithms for learning from data and making predictions. These algorithms make data-driven predictions or decisions through a model, which is built based on sample inputs.

Machine learning is used in case, when is not possible to program explicit algorithm with a solid performance.

As a science discipline, machine learning is close connected to computational statistics, which is focused on prediction-making on computers or computational systems.

Machine learning is used in data analytic for making a design for complex model and algorithm, which help with prediction. Also it is called as predictive analytics.

This analytical model helps researchers, data scientist, engineers and analysts with research of real condition of environment. This model should also produce reliable and repeatable decisions, which should be consistent.

The one of most important activities is uncovering of hidden insights through learning from historical relationships and trends in data.

### 2.3 Ambient Intelligence Environment Middleware

Autonomous unit is Python service, which communicates with hardware peripherals and sending and receiving data from external communication sources. RoboPhery unit consists of several objects. Communication objects handle sending and receiving messages from the upper layer services or other autonomous units. Interface objects handle abstraction to hardware communication at device level. Modules encapsulate individual hardware sensors and actuators. Finally the RoboPhery manager serves as central service that connects all other models within the autonomous unit.

When data are collected from sensors, it is important to data will be transferred and stored in correct state to the highest part of system, e.g. to time-series database.

Message Bus mainly take care about communication between agent in Autonomous Agent System, because data can't be easily transferred from sensor (agent) to database directly. Message Bus also can aggregate data to bigger units or make some basic transformations.

Autonomous services take care about conditions from sensors and values, which are captured. There are predefined conditions and when captured values are identical with same condition, monitoring agent send a message via message bus to reacting agent, which will performs predefined action (Fig. 4).

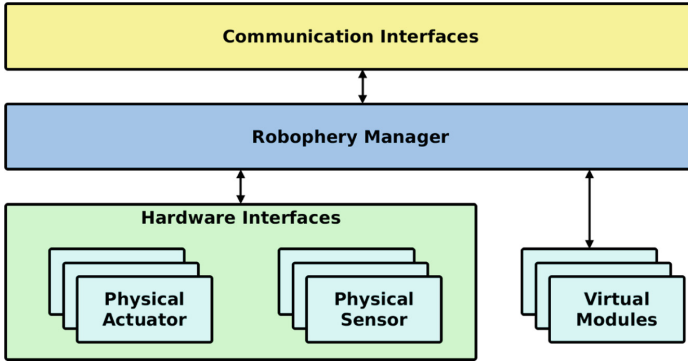


Fig. 4. Ambient Intelligence environment interactor

### 3 Sample RoboPhery Deployments

This chapter shows several use-cases of the usage of RoboPhery as hardware interfacing layer. Following Figures shows simple device configuration setups for various wired and wireless interfaces.

#### 3.1 ESP2688 ModeMCU Micro-controller

Figure 5 shows the RoboPhery manager service running on the ESP2688 ModeMCU micro-controller with multiple sensors connected to interfaces present directly on the target device. The device has low computational power and low power consumption making it suitable for remote wireless probes with several attached sensors.

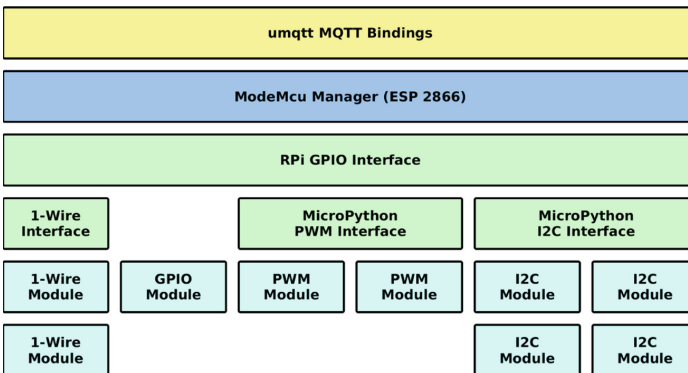


Fig. 5. ESP2688 micro-controller direct interfaces

### 3.2 Raspberry Pi 2 Device

Following figure shows RoboPhery service running on the Raspberry Pi device with MCP23008 I2C to GPIO expander boards and multiple sensors connected to individual interfaces. You can generally use multiple expander boards to support great numbers of sensor modules connected at a time (Fig. 6).

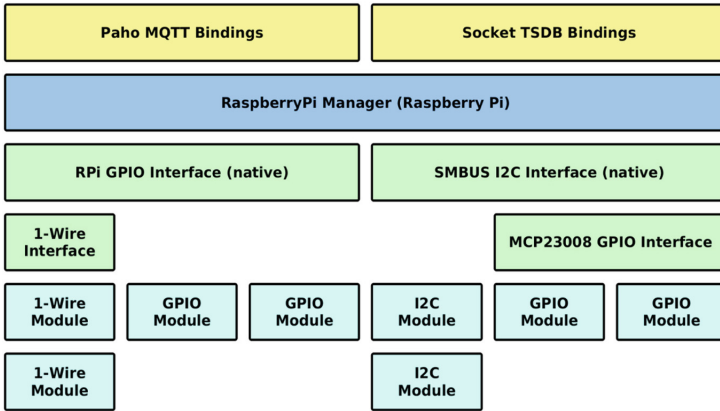


Fig. 6. Raspberry Pi computer extended interfaces

### 3.3 x86 Computer with IQRF Mesh Network

Following figure shows RoboPhery Manager service running on the x86 single board computer with USB connected to the IQRF coordinator with multiple sensor interfaces connected through a common mesh network. Individual mesh member implements an interface and can support multiple sensor modules (Fig. 7).

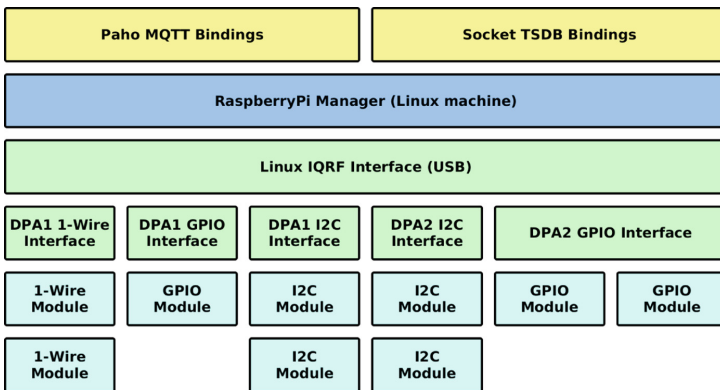


Fig. 7. x86 Computer IQRF Mesh interfaces



## 4 Conclusion

This article has shown the growing phenomena of Ambient Intelligence environment and what role it can play in daily life. We have introduced RoboPhery project for simple interaction layer for a wide variety of commonly used sensors and actuators. This layer is an integral part of any Ambient Intelligence Environment.

The main utility of the service list rests in the fact, that after little code-base changes can be operated on both computers and micro-controllers. It can reuse the sensor and actuator drivers on all platforms, thus greatly reducing the costs of migrations and reduces the vendor lock-in as you are free to use a lot of different devices available on the market. At the end the whole system is designed to be extremely modular a provides wide range of hardware interfaces and modules in the default setup and easy way to introduce any new interface or module.

The micro-controllers provide only a very limited computational power, operating memory, but it is enough to run the communication and sensoric operations. Modern single-board computers on the other hand can provide much more. They can communicate with external services, a provide enough power for other services in Ambient Intelligence stack as machine learning, time-series databases and event-driven automation. The further research is needed to explore possibilities of integrating the RoboPhery project with external services and expand the list of supported micro-controllers and devices.

**Acknowledgement.** This work and the contribution were also supported by project of specific science, Faculty of Informatics and Management, University of Hradec Kralove, Czech Republic.

## References

1. Aarts, E., Encarnao, J.L.: True Visions: The Emergence of Ambient Intelligence. Springer, Heidelberg (2006)
2. Aarts, E., Harwig, R., Schuurmans, M.: Ambient Intelligence in The Invisible Future, pp. 235–250. McGraw-Hill Inc. (2002)
3. Arribas-Ayllon, M.: Ambient Intelligence: an innovation narrative, 16 April 2017. [http://www.academia.edu/1080720/Ambient\\_Intelligence\\_an\\_innovation\\_narrative](http://www.academia.edu/1080720/Ambient_Intelligence_an_innovation_narrative)
4. Augusto, J.C., Nugent, C.D.: A new architecture for smart homes based on ADB and temporal reasoning. In: Proceedings of 2nd International Conference on Smart Homes and Health Telematic, ICOST 2004, pp. 106–113 (2005)
5. Bielikova, M., Krajcovic, T.: Ambient Intelligence within a Home Environment, 12 April 2017. [https://www.ercim.eu/publication/Ercim\\_News/enw47/bielikova.html](https://www.ercim.eu/publication/Ercim_News/enw47/bielikova.html)
6. Dunning, T., Friedman, E., Loukides, M.K., Demarest, R.: Time Series Databases: New Ways to Store and Access Data. O'Reilly Media, Sebastopol (2014)
7. Foster, I., Kesselman, C., Nick, J.M., Tuecke, S.: Grid services for distributed system integration. *Computer* **35**(6), 37–46 (2002)
8. Ha, Y., Sohn, J., Cho, Y., Yoon, H.: A robotic service framework supporting automated integration of ubiquitous sensors and devices. *Inf. Sci.* **177**, 657–679 (2007)

9. Horálek, J., Matyska, J., Stepan, J., Vancl, M., Cimler, R., Soběslav, V.: Lower layers of a cloud driven smart home system. In: Barbucha, D., Nguyen, N.T., Batubara, J. (eds.) *New Trends in Intelligent Information and Database Systems. SCI*, vol. 598, pp. 219–228. Springer, Cham (2015). doi:[10.1007/978-3-319-16211-9\\_23](https://doi.org/10.1007/978-3-319-16211-9_23)
10. Huhns, M., Singh, M.P.: Service-oriented computing: key concepts and principles. *IEEE Internet Comput.* **9**(1), 75–81 (2005)
11. Lampkin, V., Tat Leong, W., Olivera, L., Rawat, S., Subrahmanyam, N., Xiang, R.: *Building Smarter Planet Solutions with MQTT and IBM WebSphere MQ Telemetry*. An IBM Redbooks Publication (2012)
12. Li, S., Xu, L.D., Zhao, S.: The internet of things: a survey. *Inf. Syst. Front.* **17**(2), 243–259 (2015)
13. Mani Chandy, K.: *Event-Driven Applications: Costs, Benefits and Design Approaches* (2006)
14. Papazoglou, M.P.: Service -oriented computing: concepts, characteristics and directions. In: *Proceedings - 4th International Conference on Web Information Systems Engineering, WISE 2003*, pp. 3–12 (2003)
15. Papazoglou, M.P., Van Den Heuvel, W.J.: Service oriented architectures: approaches, technologies and research issues. *VLDB J.* **16**(3), 389–415 (2007)
16. Preuveneers, D., Van den Bergh, J., Wagelaar, D., Rigole, P., Clarckx, T., Berbers, Y., Coninx, K., Jonckers, V., De Bosschere, K.: Towards an extensible context ontology for ambient intelligence. In: *Ambient Intelligence: Second European Symposium, EUSAI 2004*, pp. 148–159 (2004)
17. Rui, C., Yi-bin, H., Zhang-qin, H., Jian, H.: Modeling the ambient intelligence application system: concept, software, data, and network. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **39**, 299–314 (2009)
18. Sobeslav, V., Horalek, J.: Communications and quality aspects of smart grid network design. In: Wong, W.E. (ed.) *Proceedings of the 4th International Conference on Computer Engineering and Networks*, vol. 355, pp. 1255–1262. Springer, Cham (2015). doi:[10.1007/978-3-319-11104-9\\_143](https://doi.org/10.1007/978-3-319-11104-9_143)
19. Wu, D., Rosen, D.W., Wang, L., Schaefer, D.: Cloud-based design and manufacturing: a new paradigm in digital manufacturing and design innovation. *CAD Comput. Aided Des.* **59**, 1–14 (2015)
20. Zelkha, E., Epstein, B., Birrel, S., Dodsworth, C.: From devices to “Ambient Intelligence”. *Digital Living Room Conference* (1988)