POLITECNICO DI TORINO
Repository ISTITUZIONALE

A Low Cost Open Platform for Development and Performance Evaluation of IoT and IIoT Systems

Original
A Low Cost Open Platform for Development and Performance Evaluation of IoT and IIoT Systems / Ruo Roch, M., Martina, M.. - ELETTRONICO. - (2024), pp. 155-161. (International Conference on Applications in Electronics Pervading Industry, Environment and Society, APPLEPIES 2023 Genova (Ita) 28-29 settembre 2023) [10.1007/978-3-031-48121-5_22].

Availability:
This version is available at: 11583/3005018 since: 2025-11-24T17:38:47Z

Publisher:
Springer

Published
DOI:10.1007/978-3-031-48121-5_22

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
Springer postprint/Author's Accepted Manuscript

This version of the article has been accepted for publication, after peer review (when applicable) and is subject to Springer Nature's AM terms of use, but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: http://dx.doi.org/10.1007/978-3-031-48121-5_22

(Article begins on next page)

19 November 2026

A low cost open platform for development and performance evaluation of IoT and IIoT systems

M. Ruo Roch†, M. Martina†

†Politecnico di Torino, Dipartimento di Elettronica e Telecomunicazioni

Abstract. The Internet of Things (IoT) paradigm is nowadays pervasive in a variety of applications. Distributed computing, i.e., local processing of data inside IoT nodes is mandatory to reduce power consumption and data communication cost, too. An evaluation of the needed computing power and of the hardware requirements is then as much useful as it is conducted in an early stage of the product development. The possibility to perform on-the-field experiments at an early stage is a desirable feature, too. In this paper, a low cost and open system for IoT evaluation is described. It contains basic blocks of an IoT/IIoT system, and software and hardware structures aimed to assess its performances with negligible overhead. Power consumption and timing characteristics of firmware (MCU) and hardware (FPGA) IoT nodes components can be collected and analyzed, to obtain real system requirements.

Keywords: IoT, IIoT, Ubiquitous computing, Artificial intelligence, BLE, EtherCAT

1 Introduction

The design of a new Internet of Things system is a complex problem, due to the overwhelming number of degree of freedom. In fact, this kind of application is based mainly on computation and communication tasks, which must be distributed between a variety of objects of different types [1][2]. As an example, a possible deployment is built up by a multitude of smart nodes, communicating through a low power protocol with a smart gateway. The latter, in turn, exchange data with servers located in remote data centers. This scenario is depicted in figure 1.

The represented topology is just an example, as intermediate layers can be added, or even removed. Moreover, power consumption and computing power can vary according to environmental factors (energy sources availability) and strongly influences the location of computing tasks.

Real life technology constraints must be considered, too, as the usage of off-the-shelf modules, typical in this field, can introduce far then optimal performances at the interface level between building blocks. As an example, better described later, a commonly used EtherCAT slave IC has nominal 80 Mb/s QSPI interface, but real performances could drop off to less than 1 Mb/s, due to synchronization delays related to the internal chip architecture.



Fig. 1. Typical structure of an IoT system

An early evaluation of the performances of the used modules, and of the related interfaces (wired or wireless) is then mandatory to be able to design the overall system, and to assess its feasibility for the desired system specifications.

This assessment is typically conducted in a lab, through the usage of common measurements instruments (MSOs, Logic state analyzers) on early prototypes of the nodes. But this approach requires money, and, even worse, time, as it requires the physical realization of the nodes and of the measurement bench.

A possible alternative is the usage of so-called *digital twins*, i,e, software models able to mimic the behaviour of parts of the system. Anyway, the unavailability of suitable models for some components is a not avoidable drawback of this methodology. Precision of results related to the interactions between different parts of the system is a concern, too.

Last, both approaches can not be easily adapted to true on-the-field tests, leading to a relative uncertainty in the obtained results.

Industrial IoT is a quite recent development of the IoT paradigm, sharing similar characteristics, but adding specific requirements from the point of view of reliability, safety, adn robustness. In the following, the term IoT is used to indicate IIoT, too, as the proposed solution is applicable to both fields without changes.

In this paper, a new methodology to early assess real system performances is presented. It is based on the adaptation of the VirtLAB board [3], initially developed by the authors for teaching tasks, to optimize it for IoT performance data collection and analysis. This board has the peculiarity to host both software defined measurement instruments, and the hardware on which the application runs. It means that with a reduced cost, and in a reduced space, it is already present everything needed to operate an IoT node or gateway, and to collect performance data. If needed, data analysis can be performed locally, too.

Both timing and power performances can be evaluated with the proposed system, in lab, and on-the-field. Last, in the context of Open Science and Open Innovation [4], the system is based on an open-source approach for both the hardware and the software, i.e., the system is freely available for research and development purposes, just with credits to authors.

2 Methodology and implementation

Distributed computing system implementation implies the coexistence of different computing and communication tasks on heterogeneous modules. These modules interface each other through well-known standard protocols, with known performances. But the overall performance can not be obtained just by these data. In real systems, unexpected interactions and synchronization requirements, both software and hardware, often arise, leading to real performance degradation.

Firmware and hardware changes needed to satisfy system requirements in a late design phase are a problem, too, as they can invalidate previous choices (memory footprint, clock frequencies, battery sizes, etc.)

Extensive simulations through the so-called *digital-twins* are then sometimes used, but they need extra-time, and require the availability of suitable simulation models and input data.

To overcome estimated performance uncertainties, IoT nodes are often oversized, leading to undue cost increments. An alternative approach, where feasible, is to adapt overall system specifications to the obtained ones in the deployed system.

The proposed solution is to develop a flexible and scalable system, able to output real performance data for IoT nodes and gateways. It must be used in the lab, at the beginning, but also on-the-field, for a more precise assessment.

The basic idea is that IoT nodes are typically built up by the following building blocks:

- Mixed-signal microcontrollers
- Specialized hardware accelerators
- Sensors and actuators
- Communication interfaces, wired or wireless
- Power sources

Also gateways are typically coincident with nodes, or they include a subset of the preceding building blocks.

Performance data collection must be extracted from the running hardware with minimal interaction, to not change significantly the system behaviour. This implies the usage of as fast as possible hardware methodologies, linking the measurement appliances to the application. Minimal required measurements can be summarized as such:

- Time difference measurement
- Event count
- Power consumption

Hardware and software events can be monitored just toggling a simple digital I/O line, as this is the technique which guarantees less overhead at all, up to a single clock cycle, if carefully implemented. Power consumption must be externally measured through sensing of the power supply rails.

The VirtLAB board [3] is a low cost and small size open system, developed by the authors to overcome teaching restrictions during the CoVid pandemic. It contains two sections: a user and a master ones. The user section contains a mixed-signal microcontroller and a low power FPGA, connected through a 32 bit I/O bus. Moreover, an Arduino compatible shield socket is connected to the bus, allowing to seamlessly connect existent shields. The master section is built-up by an MCU, an FPGA, volatile (DRAM) and not volatile (QSPI flash) storage. This section has access to the 32 bit user I/O bus, too, and is then able to perform time measurements on it. Current drawn by the user MCU and user FPGA can be measured in real time by the master section, too.

The hardware features of the VirtLAB board can then be easily mapped to what is required according to the preceding lists:

- Hardware modules - These blocks are built up in the user section of the VirtLAB board.
  - Mixed-signal microcontrollers - The user MCU perfectly accomplishes this task. Moreover, it is a low power device, which can be put in several sleep modes, mimicking the desired behaviour of a real IoT MCU.
  - Specialized hardware accelerators - The user FPGA can be used to implement these kinds of functions (DSP, AI, etc.).
  - Sensors and actuators - Several Arduino shields are available, with a variety of sensor and actuator types (environmental, inertial, motor control, etc.). Even if a specified device wasn't available, it would be relatively easy to design a Specialized shield to be interfaced to the VirtLAB board.
  - Communication interfaces, wired or wireless. Arduino shields are available for the more common communication standards, ranging from BLE and similar short range protocols, up to LoRa and GSM/LTE, for long range communications. Wired communications shield are available, too, for the more common industrial standards (RS232/485, CAN, EtherCAT, etc.)
  - Power sources - The VirtLAB board has an independent power supply, which derives local voltages from a simple 5V USB input.
- Measurements - Data collection and analysis is implemented in the master section of the VirtLAB board.
  - Time difference measurement - Two different solutions are available on the board. The first one, less complex but with limited performances, is implemented using the master MCU hardware timer. It is an integrated peripheral able to record the exact time at which a selected input state is changed. Up to 4 different inputs can be sampled at the same time. Time resolution is 12.5 ns, due to the internal 80 MHz clock speed, but there is a limit to the maximum frequency of events, as a DMA memory access is needed to record current time in response to an input event, leading to a bandwidth in the order of 10 MB/s. The second solution, a high performance one, is based on dedicated circuitry implemented inside the master FPGA. In this case, all the 32 bit of the I/O bus can be monitored and recorded in a high speed HyperRAM. For sake

of simplicity, resolution is held at 12.5 ns, but there is no limit to the recording speed, as data transfer toward memory will be on a high speed bus with burst transfers, guaranteeing up to 160 MB/s bandwidth.

- Event count - The two solutions implemented for time difference measurements are used for event counting, as well.
- Power consumption - Power supply lines to the user MCU and to the user FPGA have independent sense resistors and amplifiers for each rail. The outputs are sampled by dedicated ADCs channels in the master MCU, and the converted results are saved in memory through DMA accesses. Bandwidth is 350 kHz, limited not only by the current sense amplifiers, but by power decoupling capacitors, too. These are unavoidable, due to the strict requirements on power supply voltage ripple of the FPGA.

3 Usage examples

To assess the feasibility of the methodology described in the preceding section, an IIoT node has been implemented on the designed system, and the methodology has been used to extract real performances.

The first case of study has been the development of an EtherCAT slave node. The EtherCAT standard uses Ethernet technology in real time industrial applications, using specialized hardware to overcome the timing uncertainties present in the Ethernet protocol. A dedicated EtherCAT Slave Controller integrated circuit (ESC), connected to the network, is interfaced to a common microcontroller.

In our test implementation, a LAN9252 Microchip ESC is hosted on an Arduino shield, and the application microcontroller has been replaced by the VirtLAB user MCU. The implemented system is shown in figure 2

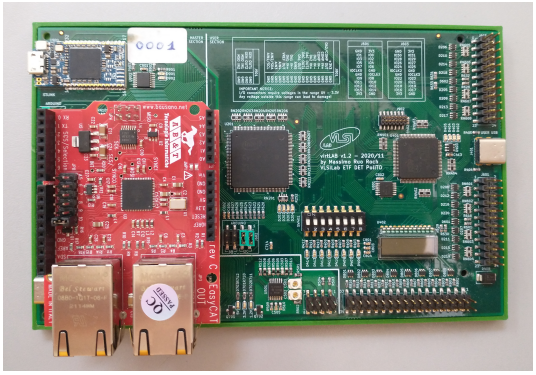


Fig. 2. VirtLAB board with EtherCAT shield installed

The ESC uses an SPI bus to communicate with the MCU, through a register based interface. The application runs on the MCU, but a FreeRTOS kernel is

used, to mitigate system timing complexity. The developed system has been used to extract accurate timing information about application behaviour. Data are sent to a PC in textual form, through a USB based virtual serial.

Latencies due to hardware and firmware interactions has been extracted, and some of the data are visible in table 1 and 2

Metrics	**Value**
Interrupt to task delay with binary semaphores	10 μs
Interrupt to task delay with direct to task notifications	8.7 μs
Interrupt to task delay with stream buffers	19.2 μs
Interrupt to task delay with polling on static shared variable	320 ns

Table 1. Latency of FreeRTOS interrupt to task, with different strategies

Metrics	**Total time**	**SPI bus time**
SPI direct register read	10 μs	15 μs
SPI indirect register read	62 μs	15 μs
SPI indirect register read with auto-increment	40 μs	15 μs

Table 2. Timings of ESC registers read, with direct to task notifications

The obtained results have been verified through the usage of standard laboratory equipment, such as oscilloscope, and multimeters.

4 Conclusions and future work

In this paper, a methodology to assess real world IoT system performances has been demonstrated. A first usage example, with quantitative data, has been shown. In the next future, other examples will be developed, to further validate and refine the designed firmware and hardware. A GUI running on the PC will be a near future development, too.

References

1. Pallavi Sethi e Smruti Sarangi. Internet of Things: Architectures, Protocols, and Applications. In: Journal of Electrical and Computer Engineering 2017 (gen. 2017), pp. 125. doi: 10.1155/2017/9324035.
2. Shivangi Vashi, Jyotsnamayee Ram, Janit Modi, Saurav Verma e Chetana Prakash. Internet of Things (IoT): A vision, architectural elements, and security issues. In: 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC). 2017, pp. 492496. doi: 10.1109/I- SMAC.2017.8058399.

3. Massimo Ruo Roch, Maurizio Martina. VirtLAB: A Low-Cost Platform for Electronics Lab Experiments. Sensors 2022, 22, 4840. https://doi.org/10.3390/s22134840
4. European Open Science Cloud (EOSC), https://eosc-portal.eu/