Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Supplementary materials for

Alireza ZIRAK, 2025. XIRAC: an optimized product-oriented near-real-time operating system with unlimited tasks and an innovative programming paradigm based on the maximum entropy method. *Front Inform Technol Electron Eng*, 26(4):568-587. https://doi.org/10.1631/FITEE.2400102

1 Foundational functions and derivations

Several essential characterizations and expressions (Schäffler, 2024; Zirak, 2023) are developed, building on foundational functions and derivations grounded in entropy principles. The proposed methods and their underlying theories are similarly derived from these principles, forming the foundation for key derivations and implications.

1.1 The additive property of information

The information learned from independent events accumulates ($I(p_1, p_2)=I(p_1)+I(p_2)$). Furthermore, the Jensen inequality asserts:

$$H(X) = E\left[\log_b(\frac{1}{p(x)})\right] \leq \log_b(E\left[\frac{1}{p(x)}\right])=\log_b(n). \tag{1}$$

The maximal entropy of $\log_b(n)$ is achieved when the system has a uniform probability distribution, signifying maximal uncertainty when all events are equiprobable.

1.2 Maximum entropy for unconstrained variables

For unconstrained random variables, i.e., with no constraints such as mean and standard deviation, the maximum entropy is achieved by a stochastic process or system with a uniform probability distribution. In other words, for an unconstrained random variable *X*, uncertainty is maximal when all possible events are equally likely, i.e., there is no preference or bias for any outcome. According to Eq. (3), if *X* takes *n* values, the entropy H_n should be maximal when all outcomes are equally likely:

$$H_n(p_1, p_2, \dots, p_n) \leq H_n\left(p_1 = \frac{1}{n}, \dots, p_n = \frac{1}{n}\right). \tag{2}$$

1.3 Entropy growth with equiprobable events

Entropy increases with the number of equiprobable events:

$$H_n\left(\frac{1}{n}, \dots, \frac{1}{n}\right) \leq H_{n+1}\left(\frac{1}{n+1}, \dots, \frac{1}{n+1}\right). \tag{3}$$

1.4 Concavity of the entropy function

Entropy is a concave function, satisfying:

$$H(\lambda p_1 + (1-\lambda)p_2) \geq \lambda H(p_1) + (1-\lambda)H(p_2). \tag{4}$$

Here, the condition $0 \leq \lambda \leq 1$ holds for all probability mass functions $p_1$and p_2.

1.5 Maximum entropy with a given mean

When considering a set $\{x_1, \ldots, x_n\}$ only with a given mean μ but an unknown variance, its distribution based on maximum entropy is characterized as:

$$\Pr(X = x_k) = C.r^{x_k} \qquad for\ k = 1, \ldots, n, \tag{5}$$

for $k = 1, \ldots, n$, as presented in Eq. (6). In this equation, the constants *r* and *C* are positive and determined based on the conditions that the sum of all probabilities must equal one, and the expected value of a random variable must be μ. Here, if each task request or job is treated as a random variable, indexed according to its repetition rate (e.g., $x_k = k$, where k is the index of each task), then:

$$C = \frac{1}{\mu-1}, r = \frac{\mu-1}{\mu}. \tag{6}$$

1.6 Maximum entropy with additional constraints

When additional parameters are considered alongside the mean μ, the maximum entropy distribution may adopt different forms such as normal, binomial, or Poisson, and these can be determined using Lagrange multipliers.

1.7 Efficiency of entropy

Entropy efficiency can be represented by:

$$\eta(X) = \frac{H}{H_{\max}} = -\sum_{i=1}^{n} \frac{p(x_i)\log_b(p(x_i))}{\log_b(n)}. \tag{7}$$

1.8 Solving probability distribution for optimal task scheduling

Given the Lagrange function:

$$\mathcal{L}(p_1, \ldots, p_n, \lambda_1, \lambda_2) = -\sum_{i=1}^{n} p_i\, ln\,(p_i) + \lambda_1(\sum_{i=1}^{n} p_i\, t_i - \mu) + \lambda_2(\sum_{i=1}^{n} p_i\, r_i - R) + \lambda_3(\sum_{i=1}^{n} p_i - 1), \tag{8}$$

we want to maximize entropy $H(T) = -\sum_{i=1}^{n} p_i \ln(p_i)$ subject to the constraints:

1 Timing constraints: $$\sum_{i=1}^{n} p_i\, t_i = \mu. \tag{9}$$

2 Resource constraints: $$\sum_{i=1}^{n} p_i\, r_i = R. \tag{10}$$

To find p_i, take the partial derivative of L with respect to p_i and set it to zero:

$$\frac{\partial \mathcal{L}}{\partial p_i} = -ln(p_i) - 1 + \lambda_1 t_i + \lambda_2 r_i = 0. \tag{11}$$

This simplifies to:

$$\ln(p_i) = -1 + \lambda_1 t_i + \lambda_2 r_i. \tag{12}$$

Exponentiating both sides gives:

$$p_i = \exp(-1) . \exp(\lambda_1 t_i + \lambda_2 r_i). \tag{13}$$

To normalize p_i (since $\sum_{i=1}^{n} p_i = 1$), we divide by the normalization factor:

$$p_i = \frac{\exp(\lambda_1 \tau_i + \lambda_2 r_i)}{\sum_{j=1}^{n} \exp(\lambda_1 \tau_j + \lambda_2 r_j)}. \tag{14}$$

2 An overview of the XIRAC commercial operating system

Designing electronic products and projects requires strategic planning to harness advancements in digital technology effectively. To avoid inefficiency and unclear goals, a practical, product-oriented approach is essential-one that leverages expertise and focuses on results.

2.1 The need for a systematic perspective

Rather than designing hardware and software components from scratch, adopting a platform-based approach enables efficient and timely development. Frameworks offering library functions and IP cores are commonly used; however, their complexity and lack of integration often hinder the creation of competitive, real-time products.

For example, platforms like Arduino and Raspberry Pi are valuable for experimentation but lack scalability and cost-effectiveness for professional, long-term applications. They struggle to support real-time, multi-functional requirements necessary for high-volume production while optimizing size, power, and integration.

2.2 Key features of competitive projects

Successful electronic systems must integrate three core components:

1 Embedded Systems: compact, processor-based boards tailored to user requirements.

2 Communication Protocols: efficient inter-module connections.

3 Graphical Interfaces: simplified control for non-technical users.

Modern designs demand real-time synergy among these elements, ensuring precise timing and functionality. All the considerations mentioned have been encapsulates into the design of XIRAC, a minimal, integrated, and optimized system resulting from over 20 years of professional experience in designing diverse products. XIRAC stands as the most powerful operating system and associated electronic circuits, which from the very first use, even without writing a single line of code, places an incredibly functional product with unique features at your fingertips. Therefore, the proper utilization of this platform can revolutionize the trajectory of your career.

2.3 Overview of the XIRAC operating system

XIRAC comprises optimized software and hardware components:

1 Software: a lightweight assembly-based RTOS for 8-bit and 32-bit microcontrollers, designed for real-time performance.

2 Hardware: modular and adaptable components, including AVR and ARM chips, suitable for various applications.

XIRAC integrates real-time functionality, seamless communication protocols, and compatibility with personal and industrial systems. For instance, in elevator systems, XIRAC facilitates synchronized operations-managing drives, monitoring cabins, and enabling remote control via robust wired and wireless communication, even in challenging environments.

Its architecture prioritizes scalability, backward compatibility, and ease of deployment. Fig.1 demonstrates XIRAC's multitasking capabilities embedded in an 8-bit processor, highlighting its ability to execute real-time tasks efficiently with minimal resources. This makes it invaluable for both educational and professional settings.

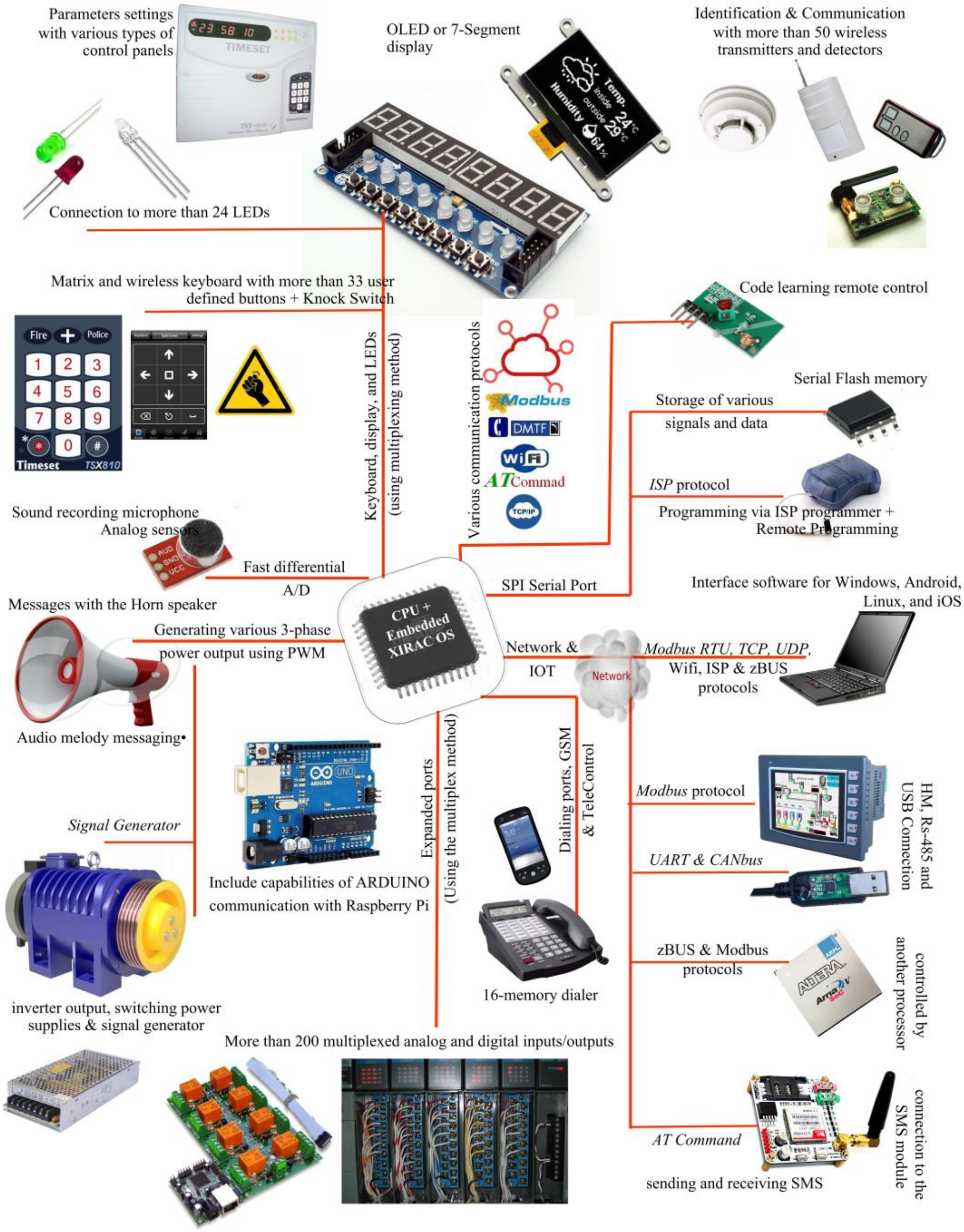

Fig. S1 Range of capabilities and communication protocols supported by the 8-bit version of the XIRAC operating system

2.4 The necessity of utilizing the XIRAC platform

Microprocessors are now integral to nearly all advanced electronic products and projects, offering significant improvements in efficiency, aesthetics, and cost. However, designing circuits with microprocessors presents several challenges that often lead to prolonged and arduous development cycles:

1 Complexity of Microprocessors: modern microcontrollers require extensive learning and substantial effort to integrate hardware and software. Even after understanding the device's architecture, writing optimized

and scalable code is a daunting task. While platforms like Arduino and Raspberry Pi simplify development, their practicality for creating competitive, cost-effective, real-time products is limited, making them more suitable for experimentation and education.

2 Rapid Technological Advancement: with the continuous evolution of microprocessors, developers face the dual challenge of adapting to new architectures while managing obsolescence of older systems. Configuring newer processors like ARM Cortex-M3 often requires intermediary layers (e.g., CMSIS, HAL), which can be highly complex.

3 Error and Noise Management: digital and electronic systems are prone to errors caused by environmental noise, such as motors, fluorescent lamps, or lightning. Addressing these issues often consumes more time than the initial design phase.

4 Broad and Evolving Product Requirements: projects frequently grow in complexity, from simple sensor monitoring to advanced capabilities like remote control, data storage, and real-time interaction via multiple platforms. Maintaining scalability while managing the software's complexity becomes overwhelming over time.

5 Standardization Gaps: the lack of integrated platforms in fields like IoT, embedded systems, and wireless sensor networks has led to fragmented, experimental research rather than product-focused solutions.

The XIRAC platform addresses these challenges as a robust RTOS (Real-Time Operating System) designed to streamline and optimize the development process. It empowers novice designers with professional capabilities while accelerating experienced developers' workflows by providing a comprehensive, feature-rich starting point.

2.5 Features of the XIRAC platform

The XIRAC system offers a unique combination of hardware and software features, enabling the development of sophisticated, scalable, and cost-effective products. Key attributes include:

1 Ease of Use: pre-configured software solutions for applications like security systems, smart home devices, motor controllers, and industrial automation are readily available. These allow users to modify or scale products with minimal technical effort.

2 Scalability and Networking: XIRAC supports large-scale networking of modules using simple two-wire communication, enabling centralized or remote control of interconnected systems.

3 Optimized Performance: the system employs assembly-based programming for maximum efficiency, with innovative methods for multitasking and hardware integration. This allows even low-cost microcontrollers to perform tasks typically requiring higher-end processors.

4 Efficient Hardware Utilization: Minimal microcontroller pins and components are required for interfacing with peripherals like displays, keyboards, and communication modules. Advanced techniques like multiplexing allow extensive functionality with limited hardware.

5 Comprehensive Communication Protocols: XIRAC supports standard industrial and networking protocols, ensuring compatibility with devices like HMIs, PLCs, and mobile phones for seamless monitoring and control.

6 Open-Source Software Tools: pre-designed interface programs and libraries in C#, QT, MATLAB, and other languages enable rapid product development.

7 Broad Applicability: XIRAC facilitates diverse applications, including industrial controllers, signal processing devices, smart home systems, and advanced scientific instruments.

These features make XIRAC a powerful platform for both educational and professional use, offering developers a versatile foundation for innovation while adhering to global technological standards.

References

Schäffler S, 2024. Mathematics of Information: Theory and Applications of Shannon-Wiener Information. Springer, Berlin, Heidelberg, Germany.

Zirak A, 2023. XIRAC-Q: a near-real-time quantum operating system scheduling structure based on Shannon information theorem. *Quantum Inform Process*, 22(11):403. https://doi.org/10.1007/s11128-023-04155-2