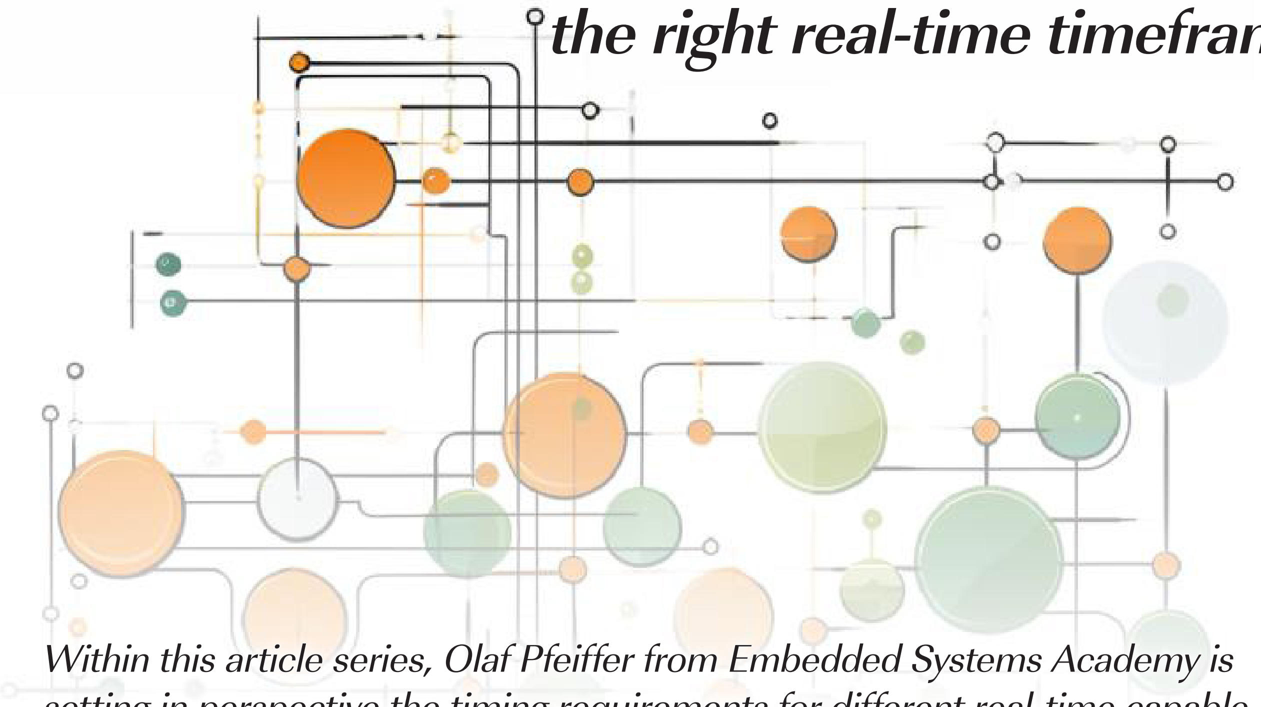
Real-time communication: Part I – Selecting the right real-time timeframe



(Source: Embedded Systems Academy)

Within this article series, Olaf Pfeiffer from Embedded Systems Academy is setting in perspective the timing requirements for different real-time capable communication systems, such as CAN, CANopen, and real-time Ethernet.

In this first article, the author takes a "step back" to first find the right perspective. Here, he reviews an application's requirement – to get an idea of the "ballpark" to be operating within. The "Part II – The demands of real-time communication systems" shows the different timeframes required by different applications and reviews what this means for the communication system used. In "Part III – The temporal dynamics of CAN-based systems," the author applies the findings from Embedded Systems Academy (EmSA) to CAN and CANopen, giving recommendations on "how to use" (configure) the communications to meet the demands found earlier. The last article "Part IV – From theory to practice: CANopen source code configuration" shows which optimization options are typically available when working with CANopen source code, here, on example of Micro CANopen Plus from EmSA.

If the required responsiveness of your system is in the area of 100 ms, then you do not need to review in detail every cause adding a delay of a single millisecond or less. To give an example, in CAN communication, collisions are resolved by prioritization. However, without collision, even the lowest-priority frame gets immediate network access. Therefore, if your system only has a busload of 50 % or less and some mechanisms are in place that no device can produce back-to-back high-priority traffic, then discussions about optimizing priorities or managing software handlers by priority may become purely theoretical, without significant practical application.

In the context of timing behavior, it is also essential to recognize that if exchanged signals require safety or security measures or both, additional metadata will be necessary to safeguard the original signal data. This may include redundant information, counters, timestamps, and various cryptographic checksums.

Selecting the right real-time timeframe

In the world of embedded systems, real-time applications occupy a crucial niche. These applications are characterized by their requirement to process inputs and produce outputs within a specific timeframe. The accuracy of the results they provide depends not only on their logical correctness but also on the precise timing of their responses. As these systems interact with the physical world, the stakes can be high, often involving human safety, product quality, or efficient system operation. Therefore, the responsiveness of these applications becomes a basic aspect of their design. However, "within a specific timeframe" can be very different depending on the application. For the rudder and thrust control of a large ship, this might be a second or more. For a high-speed sorting and packaging unit in a cookie factory, it might be single milliseconds. And these two cases already show nicely the different demands regarding safety: The "slow" commands in the ship need to be much more reliable (or safer) than those commands sorting cookies.

As you can imagine, the specific challenges of implementing real-time applications often depend on the communication channels involved. As applications tend to grow more complex and geographically distributed, it is impractical to have direct connections from the main processing unit to every input and output. Instead, many real-time systems rely on remote connections. Sensors, actuators, and other devices might be located ▷

far from the central processing unit, making some form of communication between them necessary. Often, this also means that data has to be transmitted twice within the required timeframe: Inputs from sensors to the processing unit and secondly the processing units' outputs to the actuators.

All of this brings additional challenges and considerations: Communication channels introduce delays, or potential data corruption or loss. Designers of real-time systems must now account for these factors, ensuring that the communication methods used can still meet the system's real-time requirements. In addition, these systems must now be able to handle multiple, often simultaneous, data streams and manage the prioritization of these streams based on their urgency and importance.

The increasing sophistication and requirements of real-time applications, coupled with the growing distance between the processing unit and input/output devices, have made the design of real-time systems a multi-faceted and challenging endeavor. Such a development demands a deeper understanding of communication protocols, network topologies, and error handling mechanisms. Only by addressing all these factors can we ensure that real-time systems continue to meet the stringent demands placed upon them.

Figure 1: Before diving into the design process, the first and most crucial question is determining the required timeframe for a specific application (Source: Embedded Systems Academy)

Before diving into the design process, the first and most crucial question is determining the required timeframe for a specific application. Are we talking about seconds? Hundreds of a second? Or even milliseconds? Once a system has been fully designed and developed, shortening the timeframe might not be possible, as many design decisions would have been based on the initial timeframe estimate. After you've established a desired timeframe for the real-time responsiveness of the system, the author recommends taking some extra time to review it thoroughly. Consider having your boss, customer, or partners sign off on it, as making changes to the established timeframe later on can be costly.

If your application requires that the "the entire input to output" has to be included into the calculation, then you have multiple times to add up: Processing time in the input sensor to collect the input and preparing it for transmission, transmission delay, processing time in the main processor (receiving the inputs and eventually waiting for others, processing them, and preparing for transmission to the outputs) and on the outputs the processing delay of receiving the data and actually applying it. In the following, some application examples sorted by the required response times are given.

Applications with response times beyond seconds

For applications operating in timeframes of single or multiple seconds, the systems often don't require special precautions. This is because the delay tolerance of these applications is significantly larger than the typical delays introduced by communication protocols. Interestingly, even when the control code is executed on slower non-real-time operating systems, timely operation is achievable.

Figure 2: While managing seabed operations commands may take multiple seconds to reach their destination and cause the desired action (Source: Embedded Systems Academy)

Challenges may arise if the operating system is tasked with excessive concurrent operations, but these situations are generally exceptions rather than the norm. Nevertheless, do not underestimate the consequences: Even if in your application the real-time timeframe is 1 s – what exactly will happen if that 1 s is not met? Is that just annoying – or will something get damaged – or does the data even need to be 'safe' as otherwise some serious damage or even deaths could occur? ▷

Solar tracking of solar panels: Solar panels with tracking capabilities adjust according to the sun's position. Delays of seconds to minutes are typical in this application, ensuring optimal energy capture even with occasional control delays.

HVAC systems: Heating, ventilation, and air conditioning systems often incorporate sensors to modulate temperature and air quality. While immediate adjustments are beneficial, a delay of several seconds is generally well within the acceptable range.

Mining equipment: In mining operations, large machinery such as conveyors and large-scale excavators require multiple seconds to start or stop. Given the scale, a delay of a second in system response can be acceptable, especially for non-critical adjustments. However, safety-critical functions like an emergency shut-off will have more stringent requirements.

Maritime applications: Given the relatively slow movement dynamics of large maritime vessels, a second of delay for data processing and navigation can be acceptable.

Sub-sea operations: In deep-sea systems, reliability stands as the foremost priority. While managing seabed operations – from pipeline control to equipment adjustments – commands may take multiple seconds to reach their destination and cause the desired action.

Applications with response times of 100 ms

In many scenarios, especially those centered around human-machine interaction, response times in the ballpark of 100 milliseconds are crucial. This range is rooted in the fundamental limits of human perception and reaction. When a system responds within this timeframe, the interaction feels nearly instantaneous to the user, promoting a sense of seamless control and real-time feedback. Given that the average human reaction time to visual stimuli is greater than 100 ms, systems that operate within a 100-ms timeframe are within the range to feel immediate and intuitive. To achieve these response times, one generally doesn't need to take any special measures regarding the communication channel. Even at relatively slow communication speeds like 100 kbit/s this can be reached.

Vehicle instrumentation and controls: In a variety of human-controlled vehicles, such as cars, forklifts, cranes, and agricultural vehicles, a myriad of displays and controls – from touchscreens to dials – rely on swift feedback. This ensures that the operator remains informed and in control. Sending controls via switches or joy-sticks, or receiving real-time feedback from sensors, all need to occur within this timeframe.

Industrial machine interfaces: Operators at manufacturing plants interact with complex machinery through control panels. Quick feedback is essential, ensuring the user's commands translate to machine actions almost instantly, which in turn enhances operational safety and efficiency. Where it takes longer to activate a command, some immediate visual feedback should be provided to signal the operator that the selected function is now about to be executed.

Medical equipment: Devices such as patient monitors and specific diagnostic tools require timely feedback when healthcare professionals adjust settings or input commands. This prompt response ensures both patient safety and the confidence of healthcare professionals.

Figure 3: The prompt response time of medical equipment ensures both patient safety and the confidence of healthcare professionals (Source: Embedded Systems Academy)

Applications with response times of 10 ms

For applications demanding a response time around 10 milliseconds, precision is imperative. These timings significantly surpass the boundaries of human perception, resulting in systems often responding or adjusting even before a human can register the event. Consequently, the foundational systems must operate with unparalleled efficiency and consistency. Realizing these rigorous timings demands detailed planning, balance between speed and priority, but potentially also going deep into the software layers, including drivers and firmware, that process the data. With precise optimization, these systems exhibit the ability to react promptly, reinforcing safety, preserving functionality, and assuring peak performance.

Driver assistance systems: Advanced driver assistance systems like traction control, lane-keep assist, and anti-lock brakes are paramount in delivering quick responses. These systems sense and react to instantaneous shifts in vehicle dynamics, often in situations where any delay could lead to potential accidents.

Industrial robotics: In state-of-the-art manufacturing setups, robotic arms and their allied machinery are tasked with instantaneous adjustments. Such promptness ensures meticulous precision, safeguards the sanctity of the production process, and curtails errors.

Emergency shut-off systems: In various control settings, the quick actuation of emergency shut-off systems is crucial. Whether responding to machinery malfunctions, hazardous leaks, or any unpredictable scenario, ▷

Figure 4: The slightest data exchange delay within high-precision robotics systems can lead to significant errors (Source: Embedded Systems Academy)

the swift activation of these systems can prevent significant damage, financial losses, and more importantly, protect human lives.

Applications with response times of single milliseconds

For applications that demand response times in the order of single milliseconds, the capabilities of several communication networks are stretched to their limits. Keep in mind that this is not about total throughput (typically only a few bytes are exchanged here) but get these bytes to the destination quickly. Achieving such rapid reactions requires a review of every facet of the system – from the configuration of the network to the underlying code – to be optimized. When getting into such demanding requirements, a comprehensive evaluation should be conducted to determine if the chosen communication protocol is indeed the most suitable solution or if other solutions are available to complete the tasks at hand.

High-speed motion control: In specialized industrial setups, machinery requires instantaneous adjustments based on rapid feedback loops. Such applications could involve fine-tuning motor speeds, swiftly actuating valves, or modulating high-speed actuators in real-time.

Advanced robotics: Especially prevalent in high-precision tasks, these robots might be involved in operations like placing delicate electronic components onto a PCB (printed circuit board) at accelerated speeds, where the slightest delay can lead to significant errors.

Airbag deployment: In vehicular safety systems, the time between detecting a potential crash and deploying an airbag can be mere milliseconds. Such a rapid response is crucial to ensure the safety of the vehicle's occupants, where every millisecond counts towards mitigating injury.

Real-time focus: Safety and security considerations

This article series concentrates on the timing requirements for real-time communication systems with a focus specifically on real-time performance. It is crucial to understand that considerations of safety and security are outside the review of this discussion and should be added, where required.

In safety-critical applications, redundant mechanisms are often necessary, which might require transmitting multiple CAN frames for a single command or piece of data. And, also security mechanisms require adding more data, which either makes frames longer or even adds additional frames. This means that both additional data and actions must be integrated into the established time window to ensure the system's reliability and/or security without affecting its real-time capabilities.

Conclusion Part I and outlook Part II

As we have seen in this first part of the article series, applications across various sectors have different response time requirements, ranging from seconds to mere milliseconds. The ability of a communication system to meet these needs is critical to achieving optimal performance and efficiency. However, understanding these response time requirements is only one part of the puzzle.

In the upcoming second part of this series, the author will go deeper into the specific demands placed on a communication system to meet requirements for real-time capable communication. We will explore the technical aspects that impact communication speed, latency, and arbitration, including considerations such as network architecture, bandwidth, and data processing capabilities. Furthermore, we will examine the trade-offs and compromises that must be made when selecting a communication system that strikes a balance between speed, complexity, and cost. ◀

Author

Olaf Pfeiffer
EmSA (Embedded Systems Academy)
info@esacademy.com
www.esacademy.de