The rtxd Project: Open Source Real-time for the Industrial Internet of Things (IIoT)

An open source approach to developing and assuring
real-time performance for critical IIoT applications

Scott James
President and CEO

David Warner
Director of Open Source Products and
Industrial IoT Applications

Applied Dynamics International, Inc.
3800 Stone School Road
Ann Arbor, MI 48108-2499, USA

Table of Contents

1 Summary 3
2 The Industrial Internet of Things 4
2.1 The Internet of Things 4
2.2 Consumer IoT Versus Industrial IoT 4
2.3 Growth Predictions for the IIoT 4
2.4 Key Technologies Enabling Growth of the IIoT 5
2.5 Driving Success in the IIoT 6
3 The rtxd Project 7
3.1 Overview 7
3.2 Architecture 8
3.3 Features 8
3.4 How-To Use 9
3.5 Open Source License 10
4 Project Background 11
4.1 Real-time Computing Expertise 11
4.2 Real-time Lessons Learned and Best Practices 11
4.2.1 Control System Model Frame Execution, Jitter and Troubleshooting 11
4.2.2 Efficient Use of Computation Resources 11
4.2.3 Real-time Compatible Inputs and Outputs 12
4.2.4 Test Signals and Event Handling 12
4.2.5 Synchronization Across Distributed Nodes 12
4.2.6 Reliable and Predictable Data Acquisition and Monitoring 13
4.3 The Name 13
4.4 Why Open Source? 13
5 The rtxd Project Community 15
5.1 Users 15
5.2 Contributors 15
5.3 Member Organizations 15
6 More Information 16
7 Contact Applied Dynamics 17

1 Summary

The markets for Internet of Thing (IoT) devices are forecast to experience phenomenal growth in the coming years, but not all IoT markets are created equal. Two of the most prominent IoT markets are the consumer IoT and the Industrial IoT (IIoT). IIoT devices are differentiated by much stricter requirements for timing, reliability and consistency as they are often critical links in control and safety systems. Between market demand and recent advances in technology, the IIoT is expected to accelerate an already impressive growth trajectory. With such increased demand for IIoT devices, more time and effort will be spent developing and validating these products, which in turn fuels the need for better tools and frameworks to increase development speed, quality and verifiability.

To fill this demand, the rtxd project was conceived of by ADI to share lessons learned and best practices from over twenty years of implementing critical real-time control systems for some of the world's most demanding applications. ADI believe the need for these services will only increase in the coming years and that the best way to embrace this rapidly expanding market is to share best practices through this new open source project. Figure 1 shows a basic rtxd deployment with two devices.

In the near future, ADI will be releasing the rtxd project, including source code, design documentation and examples, and will be looking for users and partner companies to join the community and help ensure safe and reliable IIoT devices for the future.

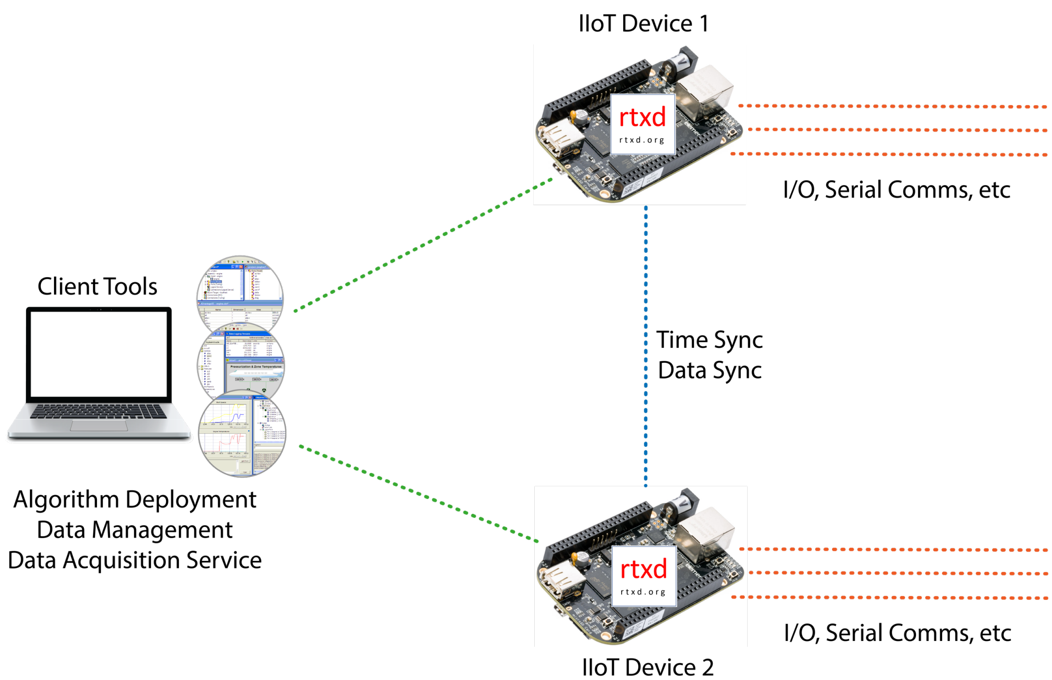

Figure 1: An rtxd deployment with two devices

2 The Industrial Internet of Things

2.1 The Internet of Things

The Internet of Things is a technological and economic concept where the world around us is peppered with small, low-cost processors with the ability to collect data through measurements and/or connectivity to installed equipment, which creates an endless spectrum of possible actions and value. The IoT has received a great deal of attention from economists and market research groups and their analysis shows near mind-blowing predictions for the economic value potential[1]. According to projections from the research company Gartner, IoT devices will outnumber humans in 2017[2]. As tech investors and Fortune 500 companies continue studying the concept of the IoT, a consensus has formed that it isn't a question of whether an IoT revolution will occur, but rather how soon and where will be the biggest market segments.

2.2 Consumer IoT Versus Industrial IoT

Consumer IoT devices are everyday objects that have some aspect being "smart" or "connected" and include wearable devices, smart appliances, "smart home" devices, vehicle installed technology, and cloud services to take advantage of this set of distributed processing. Many consumer IoT devices are designed for convenience, have loose requirements and limited impact if their performance is inconsistent. They generally undergo little testing and do not make many, if any, performance guarantees.

Industrial IoT devices, on the other hand, generally have well-defined requirements as they are often critical components in control or safety processes that could allow bad things to happen if their performance were unreliable or inconsistent. In fact, timing is often so critical for Industrial IoT devices that the consequences for not meeting timing requirements can result in a complete failure of the system, putting people and/or equipment in harm's way. Due to the stricter requirements placed on IIoT devices, they often involve much more rigorous development and testing.

Some of the major Industrial IoT markets include:

- Development, Manufacturing & Supply Chain
- Transportation & Fleet
- Utilities & Smart Grid
- Extraction and Heavy Industry

2.3 Growth Predictions for the IIoT

Initial excitement for IoT by Wall Street and tech companies was predominantly in the area of consumer IoT, but that initial enthusiasm soon cooled to some degree when cybersecurity concerns were raised and reinforced by examples of attacks on various products[3]. Excitement for the industrial version of IoT soon took over with acknowledgement that most Industrial IoT deployments would be able to take advantage of cybersecurity defense infrastructure already in place within industrial locations. Industrial deployments can leverage both low-tech cybersecurity measures such as locked doors and fences, and high-tech measures such as mature IT infrastructure, as has been deployed by most Fortune 500 companies over

[1] https://www.rcrwireless.com/20160628/opinion/reality-check-50b-iot-devices-connected-2020-beyond-hype-reality-tag10
[2] https://www.engineering.com/IOT/ArticleID/15594/IoT-Devices-to-Outnumber-Humans-in-2017.aspx
[3] http://www.digitalistmag.com/iot/2016/03/23/forget-consumer-iot-industrial-iot-will-be-the-revolution-04090972

the past decade. Although consumer IoT offers a promising growth market for tech companies, industrial applications of IoT offer lower-hanging fruit to fuel the revolution.

Today there is overabundant excitement about IIoT within both Wall Street and Silicon Valley. By having more and better data, better connectivity between equipment, and high-speed interconnections to data centers capable of analyzing and taking action, an industrial capability can be made better, with higher capacity utilization, better efficiency, and higher quality. However, one of the most surprising aspects of the IoT is the magnitude of value estimated to be provided by Industrial IoT investments. Here are some:

- Accenture: "Accenture estimates Industrial IoT could add $14.2 trillion to the global economy by 2030."[4]
- Grand View Research: "The global industrial IoT market size exceeded USD 100 billion in 2016 and is presumed to grow at a CAGR of over 25% from 2017 to 2025."[5]
- Forbes: "B2B spending on IoT technologies, apps and solutions will reach €250B ($267B) by 2020."[6]

2.4 Key Technologies Enabling Growth of the IIoT

At first glance, even with recent advances in low-cost computing, the growth predictions for IIoT may seem unreasonable. However, after considering the "perfect storm" of technologies that are maturing at the same time, the hyper-growth and ROI predictions for IIoT investments become much more reasonable. Table 1 summarizes the various impressive, fast emerging technologies that compound the benefits of an IIoT investment.

Enabling Technology	Key Benefits
Lower Cost Microprocessors and Mixed Core Architectures	Higher computational performance at lower cost with less power consumption
Real-time Linux Maturity	Lower software development tool costs, better access to skilled developers, the operating system best keeping pace with innovation
High-accuracy, Low-cost Time Synchronization	Higher accuracy data enabling greater capability to be designed into IIoT installations
Low-cost, High-speed, Time-deterministic Communications	Predictable, repeatable, distributed communications providing more responsive, higher performing IIoT deployments
Data Centric Design and Interoperability Protocols	Better interoperability between supplied equipment, better deployment workflow
Real-time Software Framework Maturity	Lessons learned from countless real-time application deployments

Table 1: IIoT Hypergrowth Enabling Technology

[4] https://www.accenture.com/us-en/insight-industrial-internet-of-things
[5] http://www.grandviewresearch.com/press-release/global-industrial-internet-of-things-iiot-market
[6] https://www.forbes.com/sites/louiscolumbus/2017/01/29/internet-of-things-market-to-reach-267b-by-2020

2.5 Driving Success in the IIoT

The computing demands placed on IIoT devices are far different from than those of consumer IoT devices, IT servers and PCs. It is generally not an issue if your smart thermostat takes a few hundred more milliseconds to change the temperature or if a website takes a few hundred more milliseconds to load in your browser, but for an IIoT device controlling a high voltage direct current converter valve stack at a utility substation, a few hundred milliseconds of delay could damage equipment and blackout electricity to an entire city[7].

Since most of the software frameworks and architectures supporting the IIoT are not nearly as mature as those for IoT and IT applications, it takes more time and effort to develop products that meet IIoT demands. The products that will be most successful are the ones that will be able to leverage the most mature frameworks to eke out the most performance from their scarce resources in the shortest development cycles.

ADI has spent over twenty years developing the real-time framework that is being used as the basis for the rtxd project. Leveraging countless lessons learned from actual real-time deployments, the rtxd project will give its users a distinct advantage over the competition when it comes to consistency, performance and reduced development times.

[7] Crolla, P. and Roscoe, A.J. and Dysko, A. and Burt, G.M. (2011) *Methodology for testing loss of mains detection algorithms for microgrids and distributed generation using real-time power hardware-in-the-loop based technique.* In: Proceedings of the IEEE 8th International Conference on Power Electronics and ECCE Asia (ICPE & ECCE), 2011. IEEE, pp. 833-838. ISBN 978-1-61284-958-4

3 The rtxd Project

3.1 Overview

The rtxd project is a set of Linux server-side codes providing out of the box functionality for time-deterministic, real-time data handling and computation, and a set of related capabilities. The rtxd server simplifies the routine tasks of developing and debugging a real-time application and allows a user to focus on their application-specific algorithms. Figure 2 shows a group of N IIoT Devices managed by a single client PC. Figure 3 shows the detail of an individual IIoT Device using rtxd to communicate with a client PC and standard I/O.

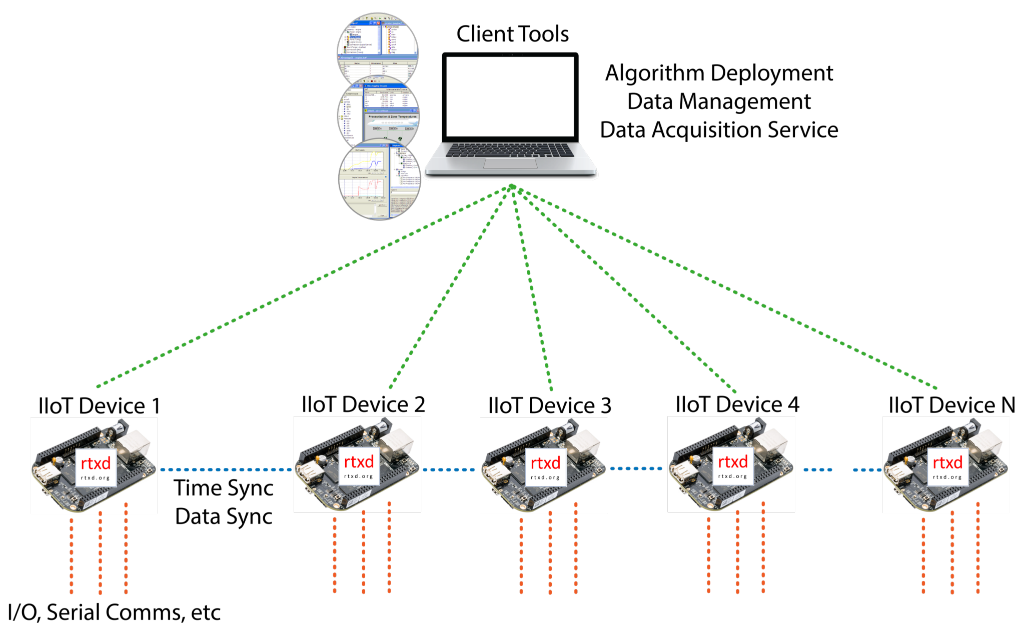

Figure 2: Communication between a Client PC and a group of N IIoT Devices running rtxd

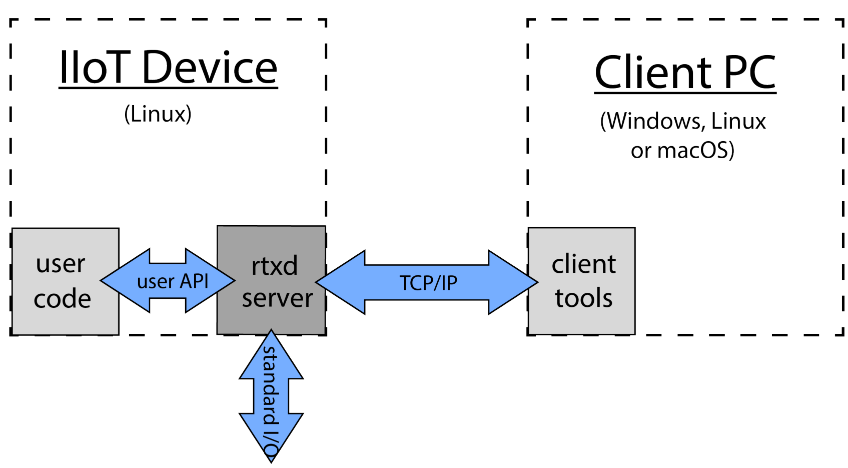

Figure 3: Communication interfaces for a single IIoT Device running rtxd

3.2 Architecture

Figure 4 shows a more detailed breakdown of the components and interfaces for an rtxd application. This architectural view adds detail to the communication channels between the Client PC and the IIoT Device, as well as within the components of the rtxd server itself.

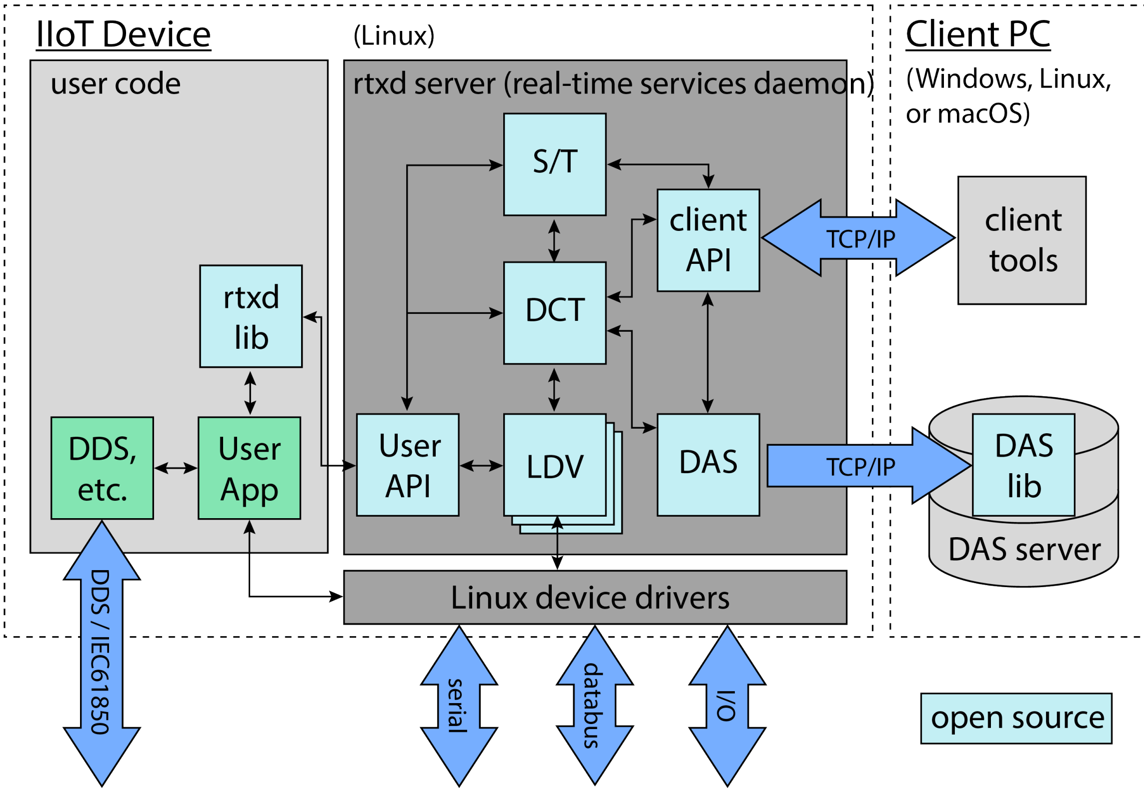

Figure 4: Architecture of an rtxd Application

3.3 Features

Here are the main features of the rtxd project:

Application / Model Scheduler / Control Loop:

The rtxd server implements its own real-time loop whose step time (frametime) is a user defined parameter, can be adjusted on the fly, and supports multi-rate execution. This real-time loop triggers the collection of performance statistics, calls to device drivers, data acquisition, user application control loop execution (through the User API) and other rtxd services.

Data Dictionaries (DCT):

DCTs are organized and managed shared memory containing data items and their properties, e.g., value, name, description, read/write permissions, data type, dimension, etc. The rtxd DCT allows user applications to focus on data-centric designs since the rtxd algorithms automatically manage the instantiation, monitoring and control of every DCT entry without the need for additional user code.

Logical Devices (LDV):

LDVs are an interface layer, translating from procedural low-level device calls to the data-centric structure of the DCT. LDVs make it easy to direct I/O and network data to and from DCT items with little to no user application code.

Schedules & Triggers (S/T):

S/Ts are time-deterministic decision engines enabling data overrides, perturbation with specific schedules such as sine waves, ramps, logic, user functions, etc., and event triggers. For example, an S/T item may be loaded to send a TCP/IP message to a specified destination if a data dictionary value exceeds a limit.

Data Acquisition Service (DAS):

DAS is a time-deterministic data logging capability allowing time-stamped data dictionary entity values to be streamed to specified TCP/IP sockets. Using the DAS requires no user application code and does not interfere with real-time performance. The DAS enables the logging frequency to be configured for each data dictionary item on the fly.

Statistics:

Every component of the rtxd project, where appropriate, is wired for time-based performance statistics that are made available through the client API. These statistics allow the user to monitor the IoT device to ensure it is operating with expected time-deterministic behavior. In addition, developers are encouraged to expose time-based performance statistics from within their user application to better characterize user application behavior.

Client API:

The client API is a TCP/IP interface that provides remote control of the rtxd server capabilities. Think of the client API as a remote control used for device management. Users can use ADI's tools or develop their own tools for interfacing to their entire set of distributed Industrial IoT devices and data infrastructure using the open API.

User API and rtxd_lib:

The user API provides the interface against which users write their device applications and is encapsulated in the rtxd_lib for easier integration.

DAS_lib:

DAS_lib is a library that makes it easier to write data collection server applications to receive data that are streamed from rtxd enabled devices.

3.4 How-To Use

One of the goals of the rtxd project is to make using the code as simple and straightforward as possible. Here are the basic steps to implementing rtxd on an IIoT project:

1. Install your preferred Linux distro on your target device

2. Install the rtxd daemon onto your Linux distro (unless you've downloaded a Linux distro that includes rtxd)
3. Write your user application(s) using whatever approach you prefer
4. Link your application(s) to the rtxd_lib API
5. Expose performance statistics, parameters, and other data to rtxd using shared memory Data Dictionaries and rtxd_lib API calls
6. Make use of those rtxd features you like, e.g. data handlers, logical devices, performance statistics, data acquisition, etc. to add time-deterministic capability to your IoT application
7. Make remote connections to your IoT device using the client API, TCP/IP interface to manage and interact with devices
8. Stream rtxd-acquired data from multiple devices to monitor performance or observe behavior of the Industrial IoT installation

3.5 Open Source License

In order to encourage organizations and individuals to adopt the rtxd project design patterns and source code, the project will have a very permissive open source license, such as the MIT License[8]. The rtxd code and design patterns may be used without royalty or restrictions whether for commercial applications or other. There will be no obligation to publish or provide changes to the code base.

[8] https://opensource.org/licenses/MIT

4 Project Background

4.1 Real-time Computing Expertise

For more than two decades, ADI has designed and implemented ultra-high-performance, time-deterministic, real-time data handling and computation facilities used to certify systems, for safety, airworthiness and seaworthiness, across the global aerospace and defense industry. In 2009, ADI predicted an emerging industrial demand for time-deterministic computing and data handling. This prompted the ADI team to define a distributed, real-time computational architecture to match the superset of capability requirements identified for this wider range of applications and larger user base. This requirements definition led to a clean-sheet redesign of the real-time server-side codes, the set of capability running in real-time on workstations, servers, and industrial computers. In May 2017, ADI released the rewritten real-time code base, the real-time executive or “rtx” codes. The rewritten and optimized rtx codes have been deployed, first in beta and now fully released, to some of industry’s most advanced real-time facilities, and those same server codes are the basis for the rtxd project.

4.2 Real-time Lessons Learned and Best Practices

The code base for the rtxd project was designed from the ground up to address key lessons learned and best practices for reliable and consistent real-time performance for critical industrial applications. Here are some of the recurring challenges faced by these systems, as well as the solutions that are supported by the rtxd project.

4.2.1 Control System Model Frame Execution, Jitter and Troubleshooting

Challenge: Critical to any real-time control loop application is the timing and reliability of the model step frame. Each iteration of the model must complete its computation within the frame time and the next frame must be started on time, not early nor late. With the number of asynchronous events that occur in a modern operating system, it can be hard to guarantee the frame execution timing. What’s worse, if there is an issue that occurs sporadically, it can be very difficult to troubleshoot what went wrong and when.

Solution: Even with a robust real-time operating system, algorithms can occasionally misbehave and cause frame overruns and/or jitter. The standard statistics built into the rtxd server provide a high level of detail to help with debugging and troubleshooting frame overruns and jitter. Specifically, the statistics include the number of overruns caused by any task, as well as detailed information about the computation time of each task, including which real-time frame caused the longest computation time. The statistics also include detailed information regarding jitter performance. Having this level of diagnostic information allows developers to quickly troubleshoot their system to understand and address the root cause of the errant behavior.

4.2.2 Efficient Use of Computation Resources

Challenge: With the increased availability of specialized processing cores, such as GPUs and DSPs, execution performance can vary dramatically based on how well the computational task is matched with the core executing the algorithms. For example, when there is a large task that can be broken down into many parallel subtasks, leveraging a GPU’s parallel processing capabilities can realize enormous performance benefits over running the algorithm on a CPU core.

Solution: The rtxd project was designed to allow different algorithms to be targeted to different processing cores. There is currently robust support for allocating specific algorithms to CPU cores. GPU cores can be targeted in some systems and future support is planned for taking advantage of DSP core capabilities.

4.2.3 Real-time Compatible Inputs and Outputs

Challenge: Inputs and outputs, especially when complex asynchronous communications are involved, can wreak havoc on real-time performance if not appropriately designed and implemented. There are many approaches to address these issues, including offloading complex asynchronous processing to a dedicated core and designing drivers in such a way that they are non-blocking and pre-emptible, but all of the approaches require a deep understanding of their real-time impact as well as careful implementation and verification.

Solution: The Logical Devices for rtxd are specifically designed in a way that minimizes negative impact on real-time performance. Therefore, when a user application links an input or an output to a Logical Device interface, it can be assured that any I/O actions will place a consistent and predictable load on real-time performance, saving considerable time and effort for these interfaces. When the rtxd project is released, see the documentation to understand which standard I/O are currently supported by Logical Devices.

4.2.4 Test Signals and Event Handling

Challenge: The simple act of generating test signals and handling real-time events can, if not properly implemented, negatively affect the real-time performance of a system. As applications grow in criticality and/or complexity, the testing requirements grow exponentially, which can easily impact the ability of a system to meet its real-time performance requirements.

Solution: The Schedules and Triggers engine built-into the rtxd server provide the seamless ability to schedule test events and to act on user defined conditions. This functionality can be leveraged to simplify the implementation of test cases and to immediately take action if any parameter is outside of a defined specification. Triggers and schedules may also be used to implement continuous Built-In-Test (BIT) capability, including real-time signal range checks, redundant input signal divergence, and complex event notification triggers.

4.2.5 Synchronization Across Distributed Nodes

Challenge: Synchronizing distributed nodes of a system can be extremely challenging due to differences in local clocks, asynchronous I/O and management of node firmware version and control.

Solution: The rtxd project was designed to work with systems that use PTP (IEEE 1588) for clock synchronization and the built-in interfaces for Data Dictionaries, Logical Devices and Model Control greatly simplify implementation and management of distributed systems.

4.2.6 Reliable and Predictable Data Acquisition and Monitoring

Challenge: As is the case with the Observer Effect in physics[9], the act of monitoring data in an IIoT device can easily alter the system itself. Special care must be taken to ensure that any data acquisition algorithms are implemented in such a way that they have little to no impact on the real-time performance of the system and that their behavior is always consistent and predictable.

Solution: With the built-in DAS capabilities, the rtxd server provides an easy way to monitor and stream Data Dictionary items and statistics for any application without compromising consistent and predictable real-time performance.

4.3 The Name

The rtxd name was created by taking the name of the real-time executive, rtx, and adding a “d” on the end to denote a daemon or service in Linux/Unix nomenclature. See Figure 5 for a graphical representation.

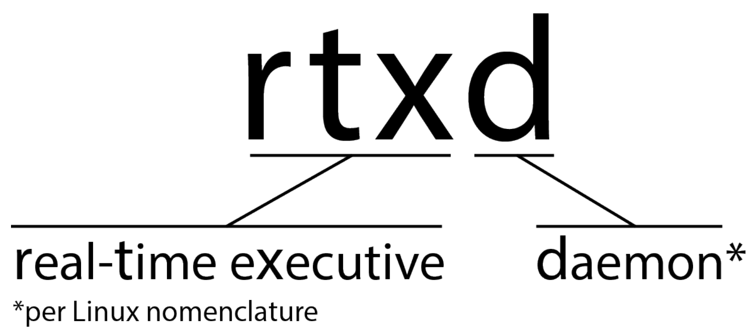

Figure 5: Etymology of the rtxd name

4.4 Why Open Source?

The rtxd project is being launched for similar reasons to any impactful open source project:

- Accelerate product innovation by collaborating with a global community
- Reduce obsolescence risk by sharing the expertise and ownership of the technology with the community
- Increase adoption of the technology by reducing barriers to entry
- Maximize technology reuse by lowering the individual cost/effort of implementation
- Share the future development and support burden by leveraging the community resources
- The wider adoption of quality real-time server code can improve the safety and reliability of critical applications and has the potential to save lives and improve the quality of life for everyone

[9] https://en.wikipedia.org/wiki/Observer_effect_(physics)

Additionally:

- ADI strongly believe there is a far wider demand for this category of computationally optimized capability, and if they do not open source a code base with this functionality then someone else would
- ADI will benefit from a larger adoption of the rtxd approach as some percentage of users will benefit from purchasing ADI's engineering services support and/or licensing for ADI's ADvantage Framework Tools

5 The rtxd Project Community

5.1 Users

A user of the rtxd project is anyone that takes advantage of the rtxd design pattern or open source code in their application. Users are encouraged to benefit from the countless lessons learned that are designed into the rtxd project. Users are also encouraged to become contributors, even if just by submitting bug fixes or feature requests to the community.

5.2 Contributors

Contributors are people that contribute their time or resources to improving the rtxd project for the community. Contributions can range from simply reviewing code or filing bug reports to submitting new features and functionality to the code base.

5.3 Member Organizations

Since the rtxd project is chartered to be completely independent of any one company, it will be owned by its member organizations. These member organizations will ensure the future support and availability of the rtxd project code by contributing an annual membership fee. The details of the rtxd organization structure and membership will be detailed after the initial public release of the rtxd project.

6 More Information

Anyone who is interested in learning more about the rtxd project is encouraged to join the mailing list or follow the project on social media. For more information, visit the project website at www.rtxd.org.

7 Contact Applied Dynamics

USA
Applied Dynamics International, Inc.
3800 Stone School Road
Ann Arbor, MI 48108-2499, USA
Tel: 734.973.1300
Fax: 734.668.0012
Email: adinfo@adi.com

USA
Mr. Alan Strech
Applied Dynamics International, Inc.
32201 Crystalaire Drive
Llano, CA 93544-1240, USA
Ph: (661) 944-1969
Email: alan@adi.com

CHINA (PRC)
Beijing
Ensky Technology Co., Ltd.
Mr. Wang Liwu
Room 1031, Section C, Chaowai SOHO
No.6 B, Chaowai Street
Chaoyang District, Beijing
100020, China
Tel: 010-59009377, 59009378;
Mobile: 13601227682
Fax: 010-59002883
Email: wang.liwu@ensky.com.cn

JAPAN
Kyokuto Boeki Kaisha Ltd.
Mr. Yuichi Wada
New Otemachi Bldg., 2-1, Otemachi 2-Chome,
Chiyoda-ku, Tokyo 100-0004, Japan
Tel: 81.3.3244.3823
Fax: 81.3.3246.2765
Email: wada@kbk.co.jp

UNITED KINGDOM
Applied Dynamics International, Ltd.
The Conference Centre
Chapel Street
Melbourne, Derbyshire DE73 8EH UK
Tel: 44(0)1536.410077
Email: adiukinfo@adi.com

FRANCE
ASC – Applied Software & Consulting
M. Gilles Derio
112, av Kleber
75784 Paris Cedex 16, France
Tel: 33.0.1.47.55.74.00
Fax: 33.0.1.64.22.94.13
Email: gilles.derio@asc-sas.com

ITALY
Otopos
Sig. Gianfranco Cattadori
Via Tetto Nuovo 10/G
10025 Pino Torinese, Torino, Italy
Tel: 39.011.811.1145
Mobile: 39.335.60 75 110
Fax: 39.02.700446492
Email: g.cattadori@otopos.it

KOREA
Anawell Corporation
Ms. Y.E. Park
Shinil Uto Vill Rm 711
735-11 Yuksam-Dong, Kangnam-Ku
Seoul 135-080, Korea
Tel: 82-2-554-2173,
Fax: 82-2-554-2175
Email: yepark@anawell.com