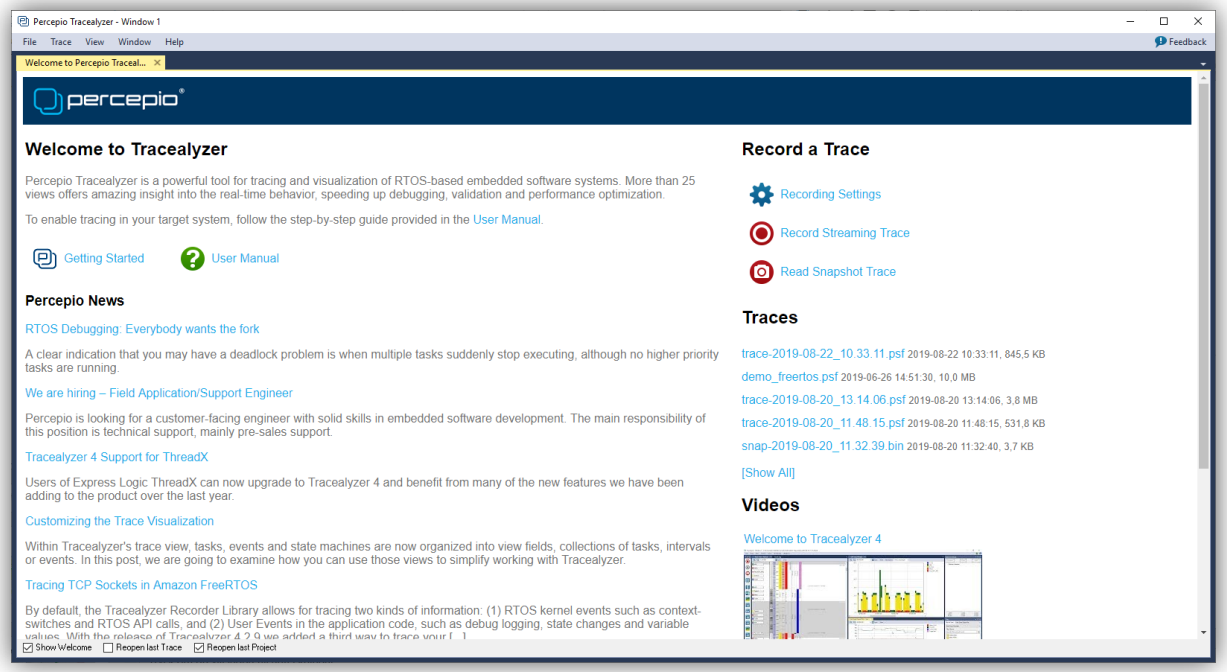
# Benefits of Tracealyzer vs. Trace Compass

Johan Kraft, Percepio AB

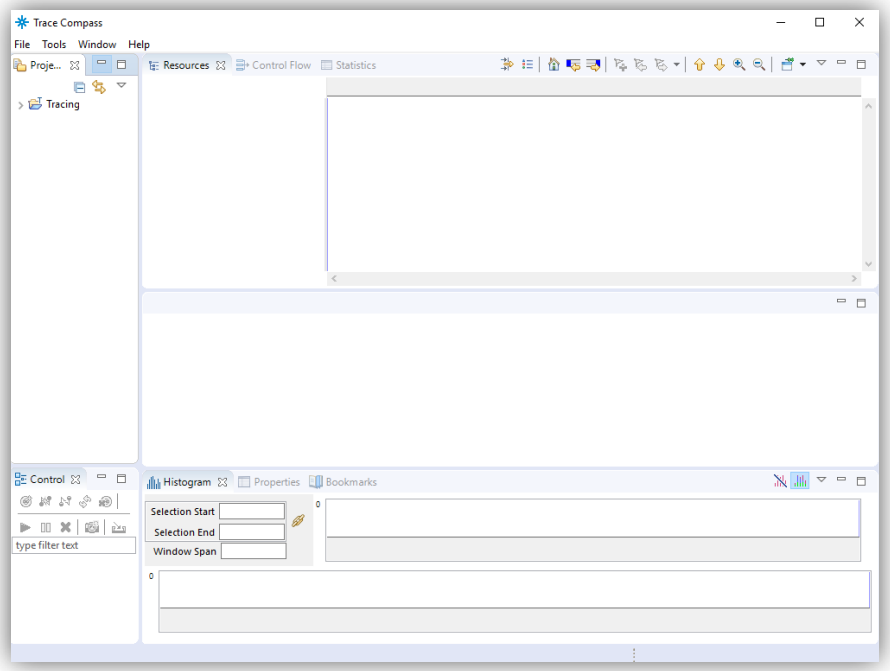
Sept. 2, 2019

# Tracealyzer v4.3.5




Copyright: Percepio AB
License: Commercial, Proprietary
Development start: 2004/2009

# Trace Compass v5.0.0




Copyright: ”Ericsson and others”
License: Eclipse Public License v1.0 (Open Source)
Development start: 2013

percepio®

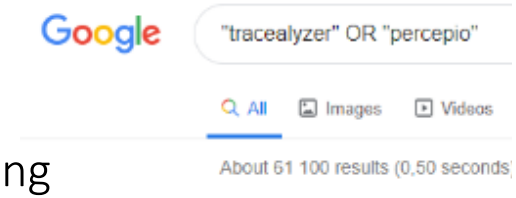
# High Level Comparison

# Tracealyzer Product Benefits

- Easier workflow, designed for *application* developers
- 30+ graphical views, lightning fast and with better drill-down
- True cross-platform analysis capability, not just focusing on Linux
- A portable tracing library for supporting other RTOSes
- Cloud-based device monitoring
  - Percepio Device Firmware Monitor, see https://percepio.com/dfm
- Eclipse interoperability
  - https://marketplace.eclipse.org/content/percepio-trace-exporter

# Popularity and Trends

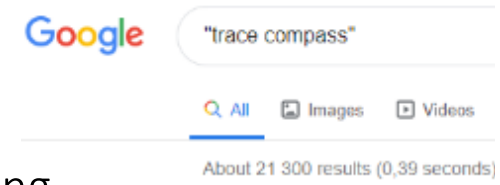
## Tracealyzer




Google hits: 61 100

Google Trends: Increasing

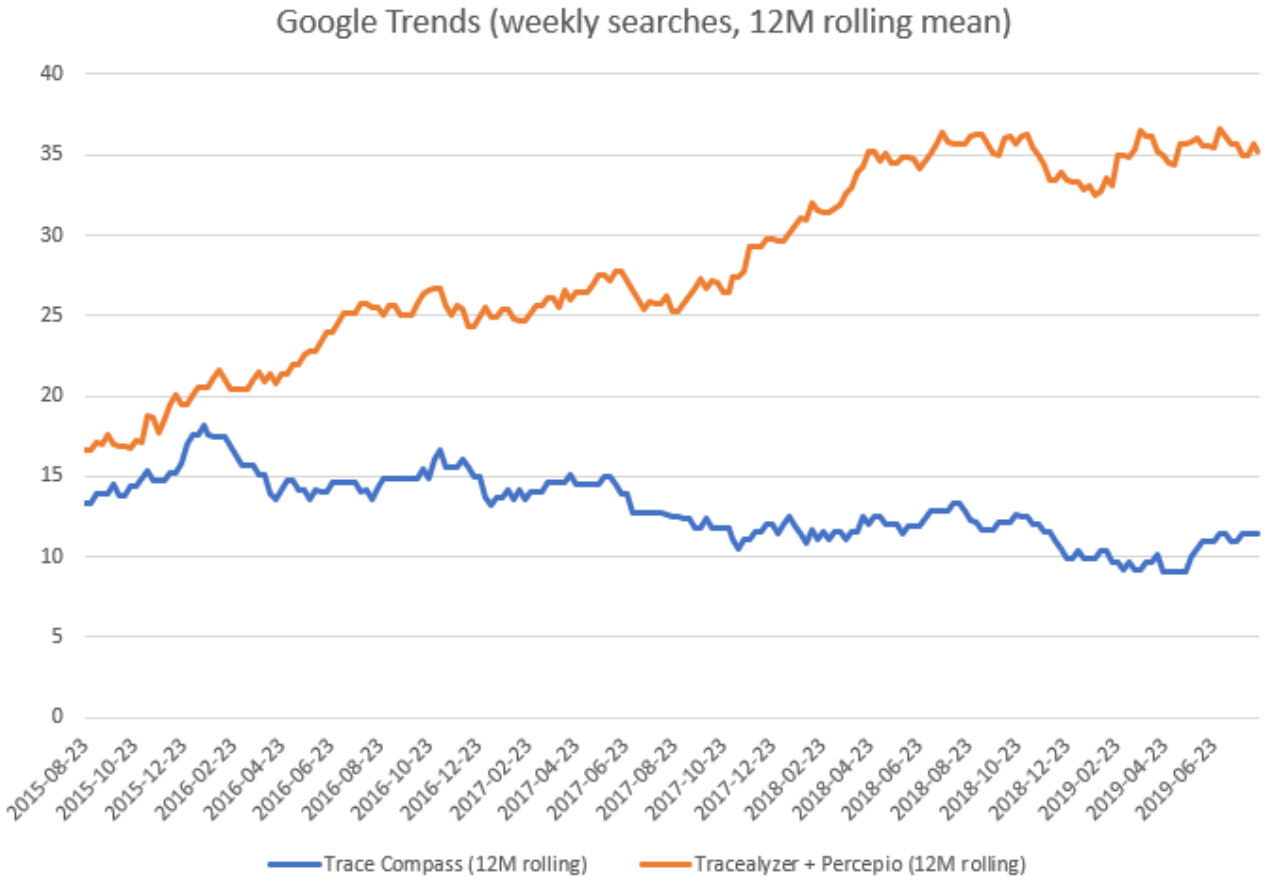
## Trace Compass




Google hits: 21 300

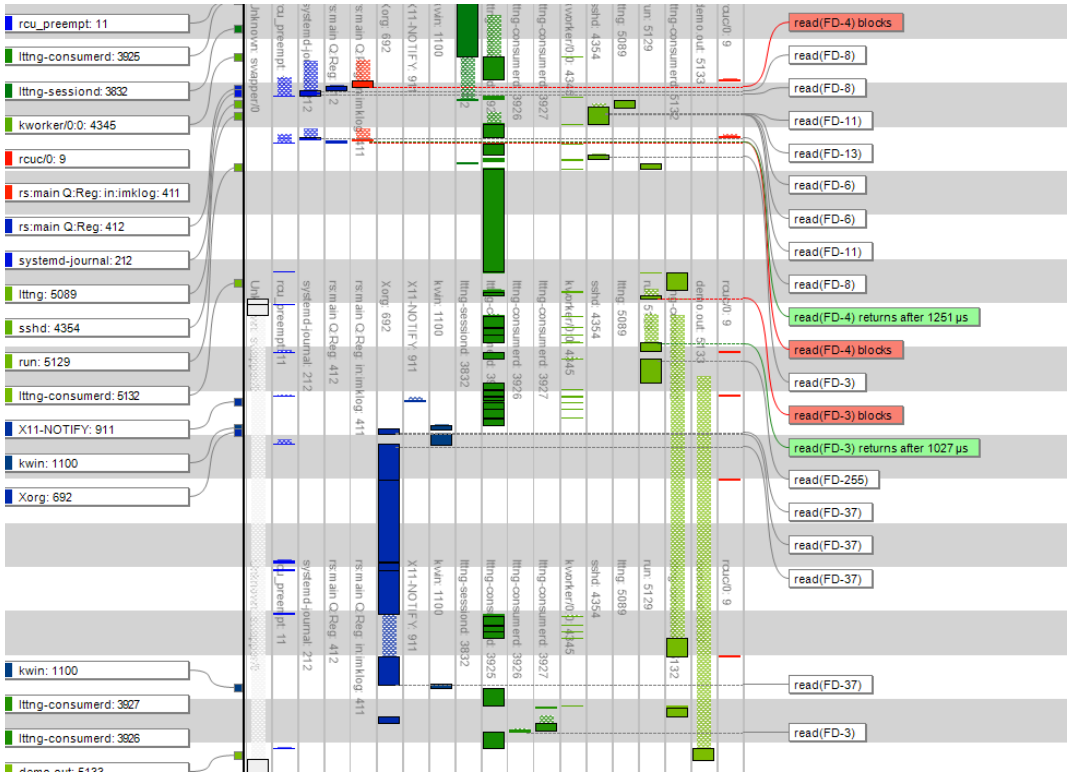
Google Trends: Decreasing

(This despite being a free open source tool…)

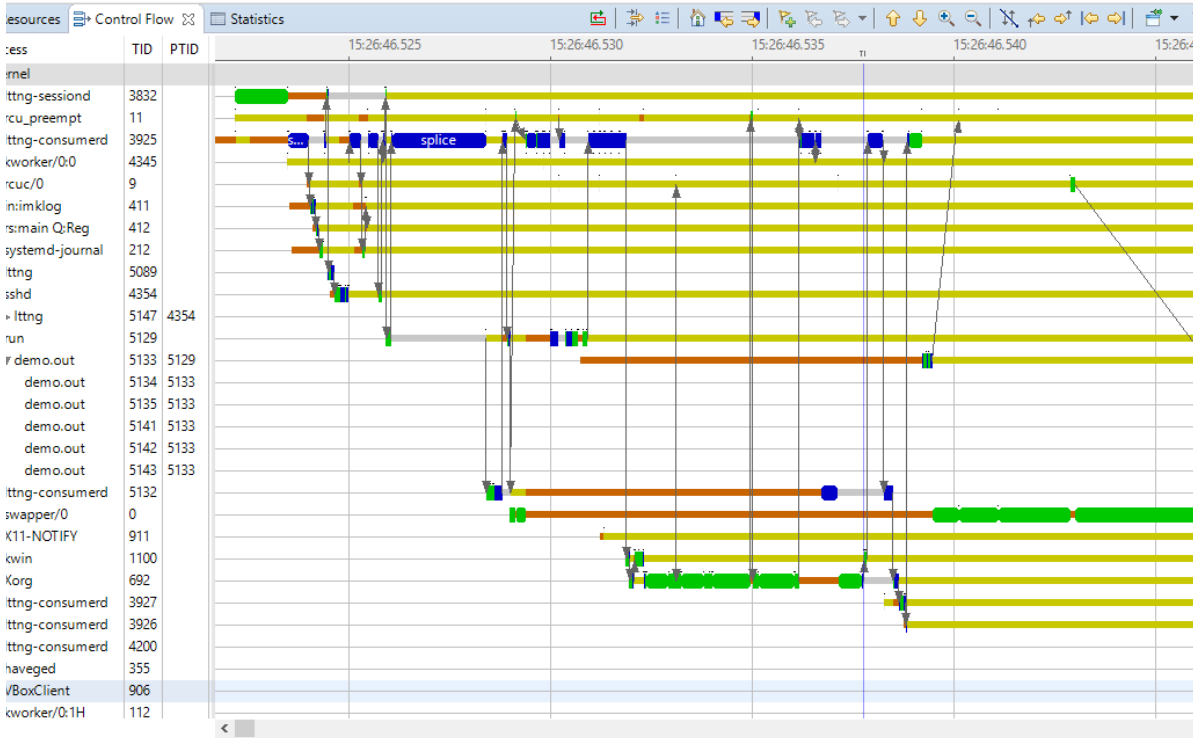


# Features and User Experience

# LTTng Kernel Trace (20 ms, same data)

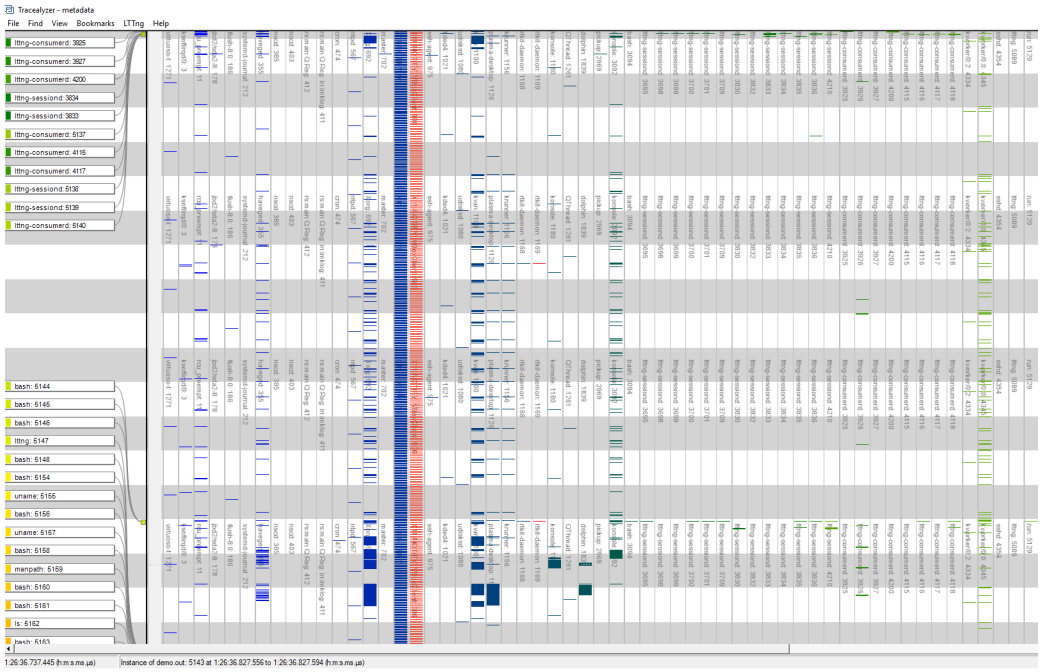



Tracealyzer for Linux (old version)




Trace Compass v5.0.0

# When Zoomed Out… (8 s, same data)





Not much visible…

Tracealyzer for Linux (old version)

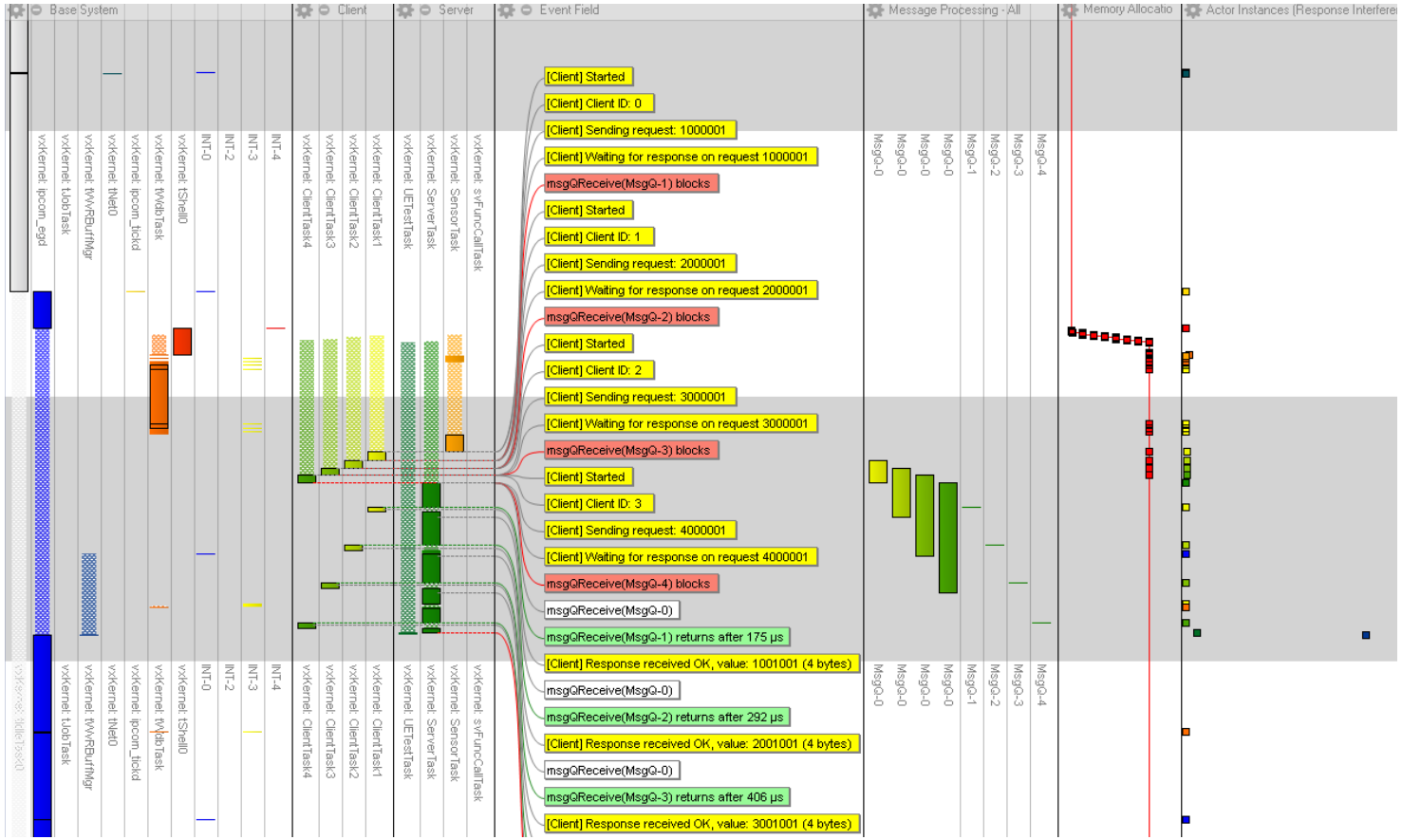
Trace Compass v5.0.0

# Examples of Unique Tracealyzer Features

- Superior Trace View
- More Application-focused Views
  - Communication Flow
  - User Event Signal Plot
  - State Machine Graph
- “Actor Instance” Awareness for Profiling of Real-Time Tasks
- Other things…
  - More extensive and user-friendly documentation, blog examples, etc.
  - Antialiasing etc. for better graphics
  - Parallel views always in sync (minimal update delay)
  - …

*All features are designed to be cross-platform.*
*Not just Linux, also VxWorks and other RTOSes*

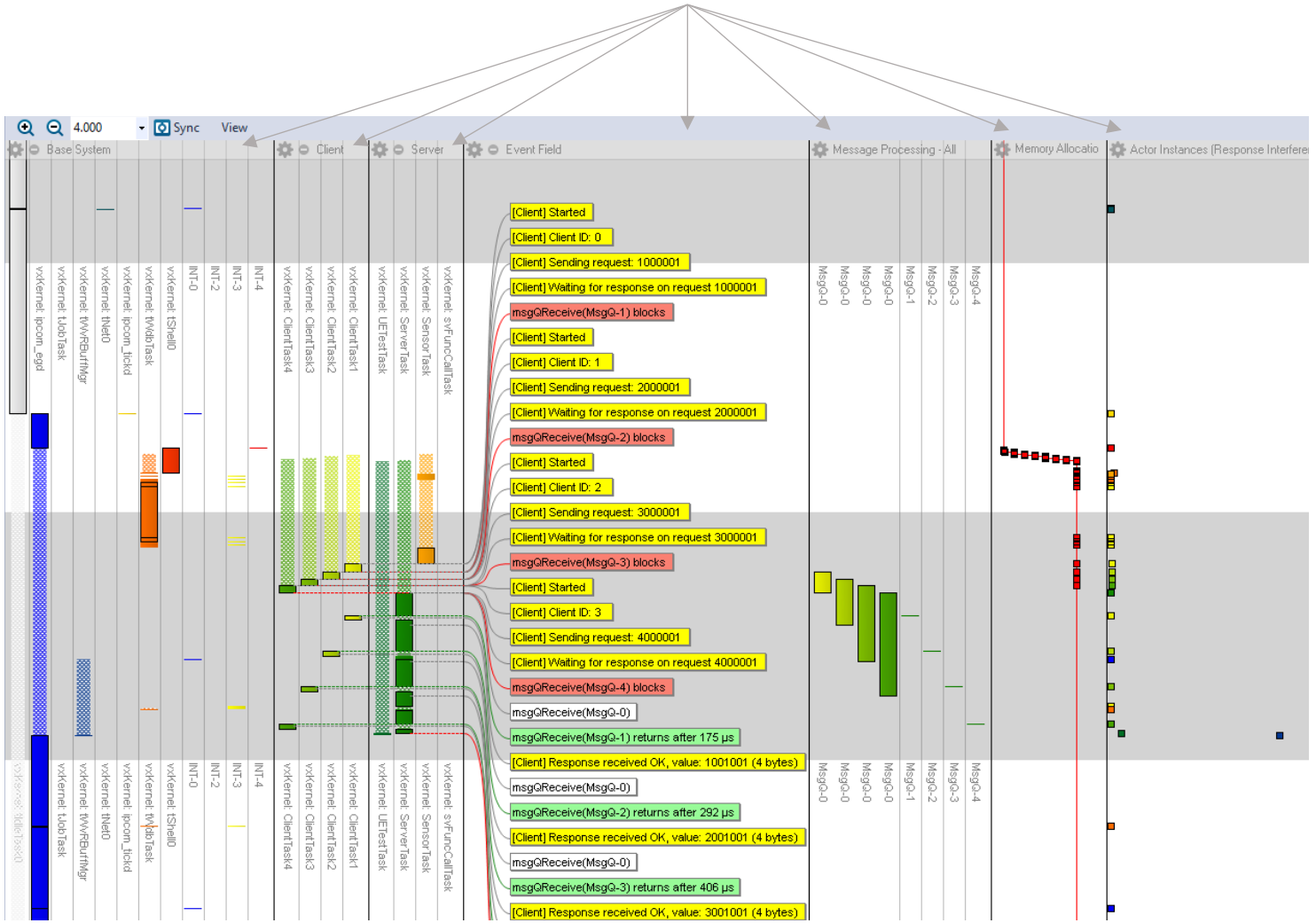
# Unique Features – Superior Trace View




- Configurable view fields
- Vertical or horizontal orientation
- Events shown as floating labels
  - API Calls
  - Task State Changes
  - User Events
  - …
- Awareness of blocking/timeouts
- Scales to long traces

(Vertical timeline in this example)

# View Fields




## View Field Types:

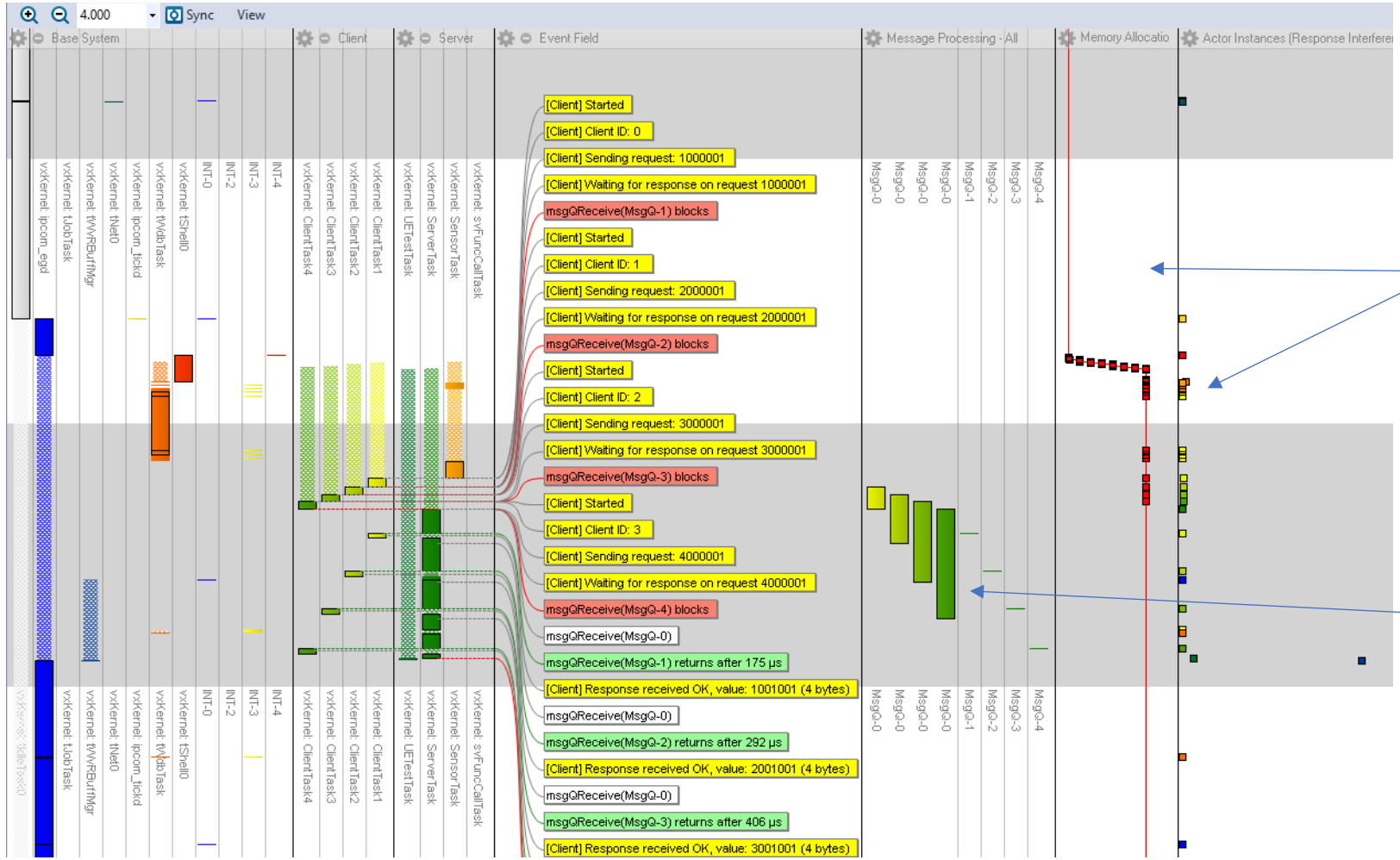
- Scheduling field
- Event field
- Interval field
- Metrics field

Scheduling Fields

Events Field

Interval Field

Metrics Fields




## Metrics Field

Examples shown here:

- Heap Memory Usage (malloc/free)
- “Response Interference”

Today 16 metrics available

## Interval Field

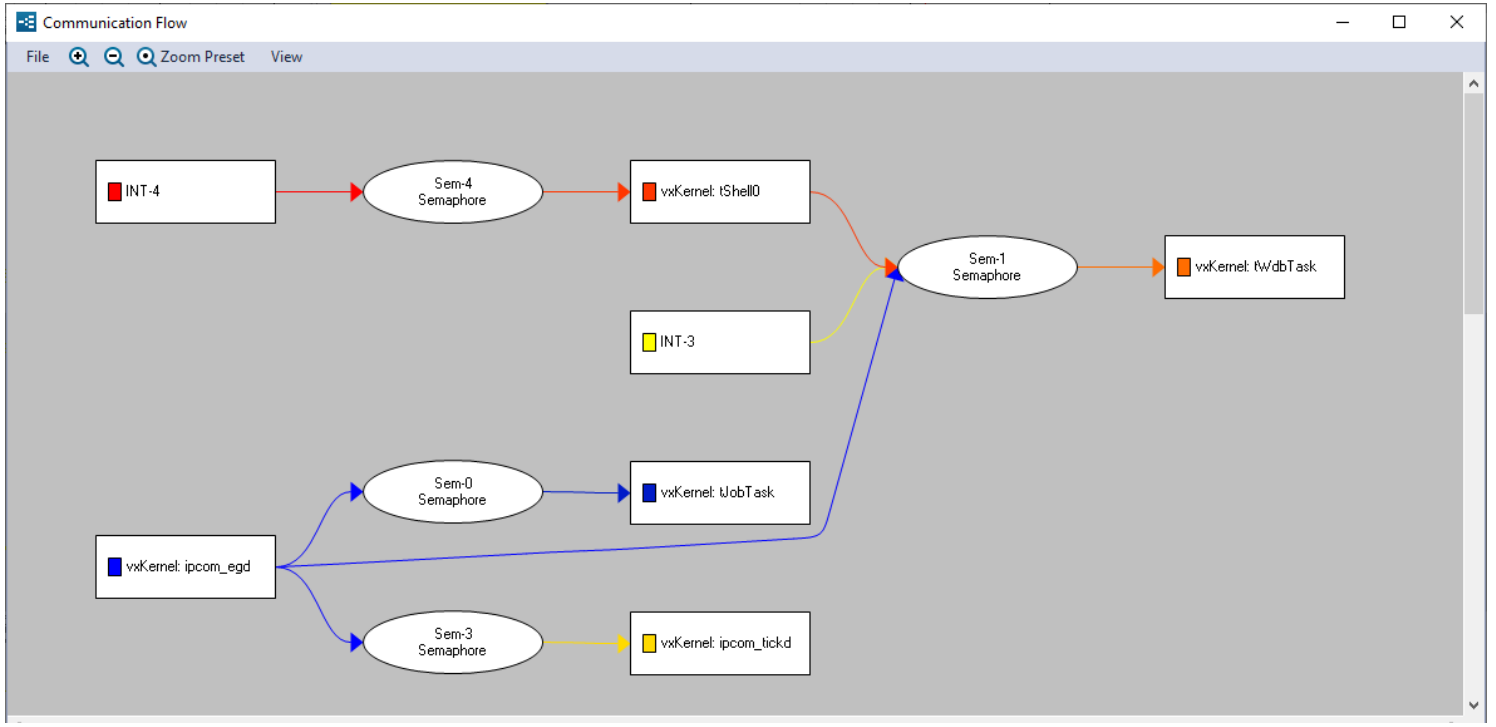
Example shown here:
MsgQ messages (from msgQSend to matching msgQReceive)

Several other predefined data sets available, and also **custom intervals**.

Can also show **states**.

percepio®

# Unique Features – Communication Flow

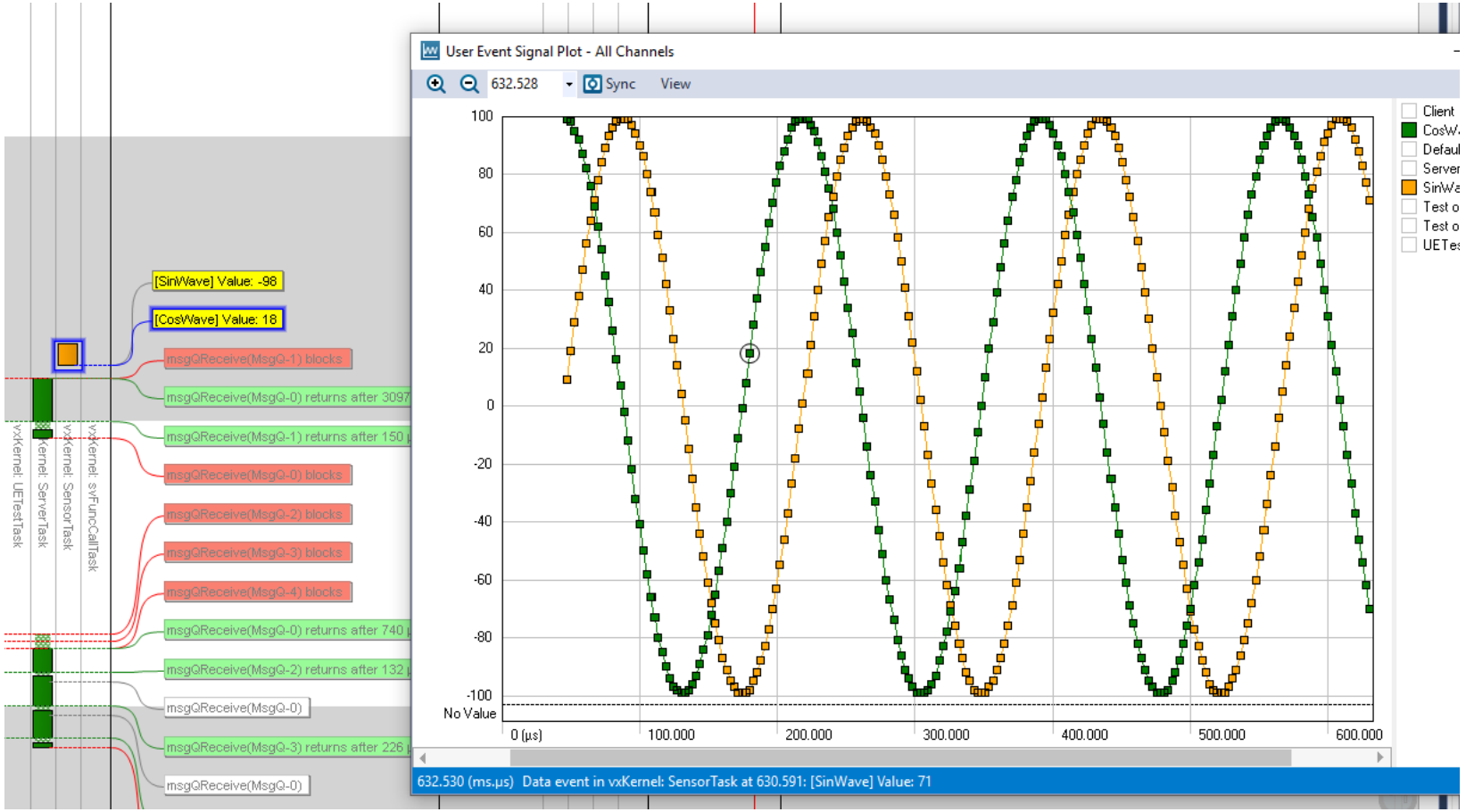



A dependency graph showing how threads/tasks/ISRs are interacting via kernel services.

Generated from the kernel trace data, based on the whole trace or a selection.

Double-click to show matching events

# Unique Features – User Event Signal Plot

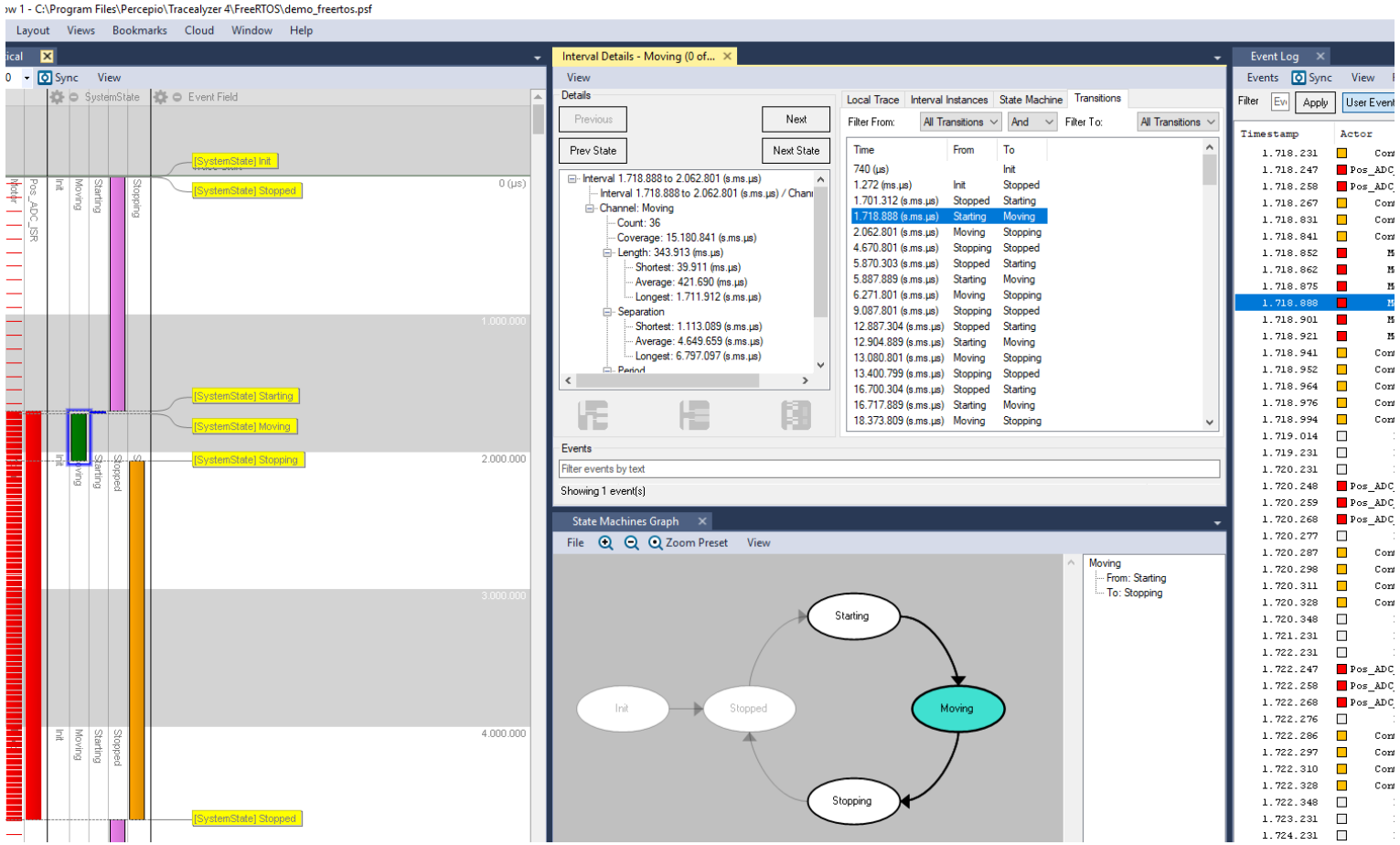



Generic data plot for User Events

- System inputs and outputs
- Control loop signals
- Other variable values
- Performance counters
- …

Each data point is linked to the trace view.

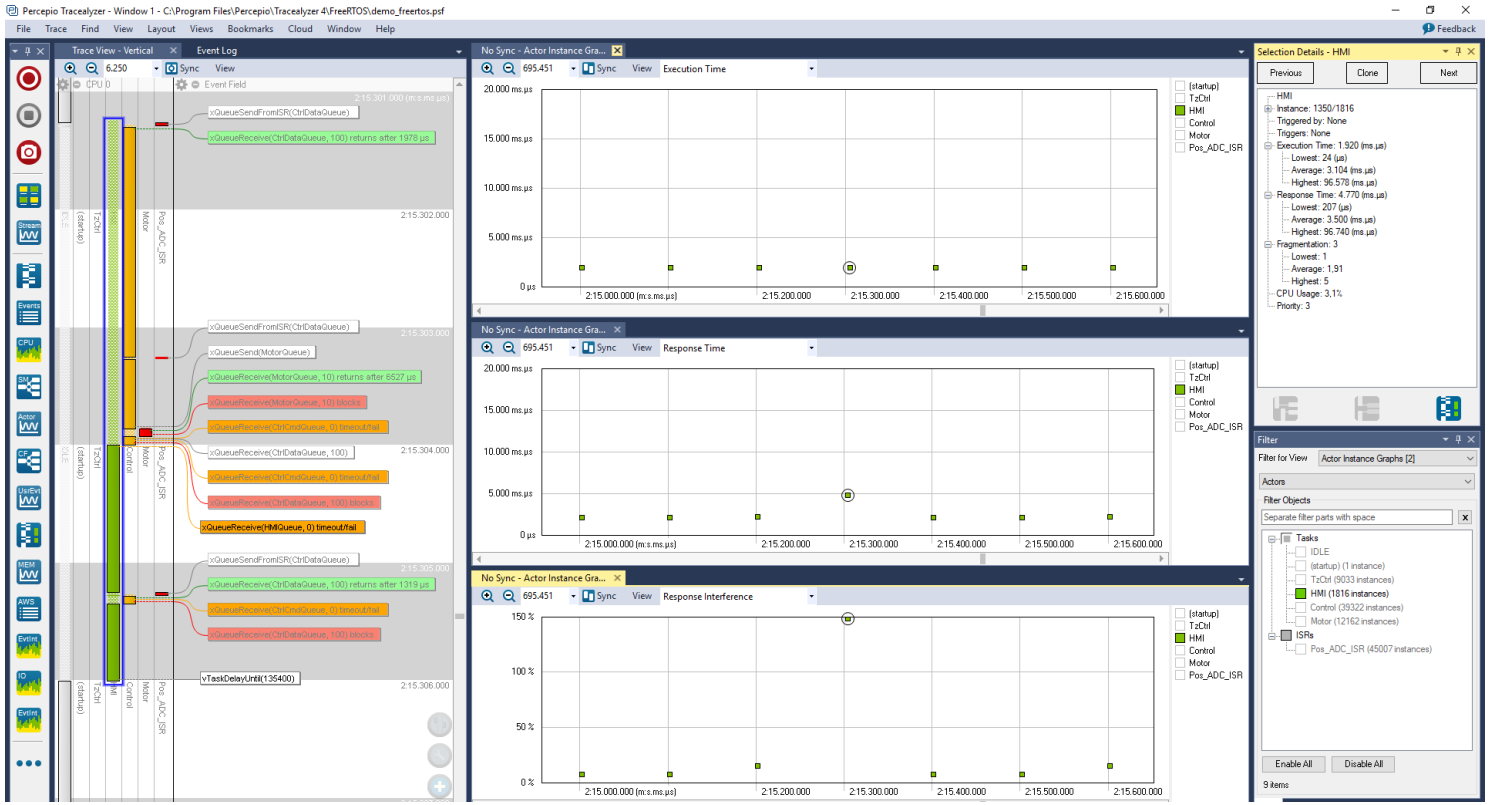
# Unique Features – State Machine Graph




State transition diagrams can be generated from “state events” (logged software variables or hardware states)

States also shown in trace timeline

# Unique Features – “Actor Instance” Awareness




Awareness of task jobs (ready to finished) for better profiling of real-time tasks.

Instance-level metrics:

- Execution time
- Response time
- Fragmentation
- Periodicity
- Separation

Plots over time, per instance

Report generation