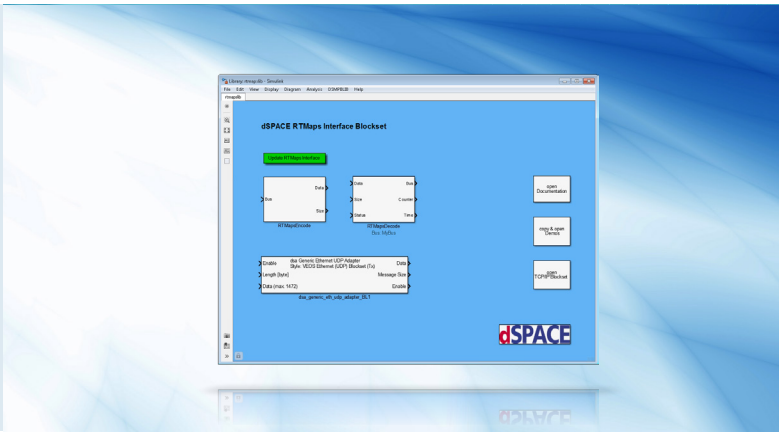
# RTMaps Interface Blockset

## Enabling full-stack prototyping across multisensor applications and hard real-time controllers

## Highlights

- Simulink® blockset for low-latency communication and clock synchronization between dSPACE platforms and RTMaps
- Recording and playback of time-correlated data
- Easy development of applications for ADAS and automated driving




### Application Areas

The conventional vehicle software stack is at its limits, as hard real-time controller-based prototyping is no longer sufficient for advanced driving assistance systems and autonomous driving (ADAS/AD). As a result, two prototyping worlds arose, where one is dedicated for perception and data fusion, and the other is aimed at hard real-time execution.

Typically, data processing and fusion algorithms are implemented in a high-level programming language such as C++, while the development of application functions rests on the model-based design approach using MATLAB®/Simulink® and dSPACE real-time systems. RTMaps from Intempora is an established tool with an unparalleled performance for prototyping C++-based multisensor applications using block diagrams. In addition, RTMaps provides capabilities to precisely time-stamp, record and play back data.

The dSPACE RTMaps Interface Blockset connects dSPACE platforms, such as dSPACE real-time systems, or VEOS with RTMaps via UDP/IP.

### Key Benefits

The dSPACE RTMaps Interface Blockset provides a bidirectional, low-latency communication via UDP/IP between RTMaps from Intempora and dSPACE real-time systems or the PC-based simulation platform dSPACE VEOS. In just a few steps, developers can connect their applications in Simulink to the appropriate communication blocks via signal buses and establish data connections to multisensor applications in RTMaps. The required description of the data to be transmitted or received is generated as an XML file from the corresponding signal buses in Simulink at the push of a button. In addition, the blockset allows the clock in RTMaps to be synchronized with the simulation time or the real-time clock on dSPACE platforms. This makes it possible to capture, record, and precisely time-correlate vehicle bus on dSPACE MicroAutoBox II/III and camera data in RTMaps. Another use case is the open-loop (data replay) test of ECUs for image processing. Here, RTMaps on a PC plays back time-correlated video and bus data while the bus signals are sent to a hardware-in-the-loop (HIL) system. This ensures a low-jitter restbus simulation.

The RTMaps Interface Blockset establishes a powerful connection between RTMaps and dSPACE platforms and facilitates the development and testing of perception and application algorithms for ADAS and automated driving.

## Functionality Overview

| Functionality | Description |
|---|---|
| General | ■ Low-latency communication between RTMaps from Intempora and dSPACE platforms via UDP/IP<br>■ Simulink blocks for encoding/decoding data from Simulink buses into byte streams and vice versa<br>■ High-level communication protocol with clock synchronization regarding simulation time or (with MicroAutoBox II/III) the real-time clock<br>■ Detection of packet losses and data corruption by means of counter and checksum mechanisms |
| Ethernet communication | ■ Blockset for TCP/IP- and UDP/IP-based communication with MicroAutoBox II/III<br>■ Up to 16 sockets or connections with combined sending and receiving queues<br>■ Multi-rate modeling and data pockets of variable length |

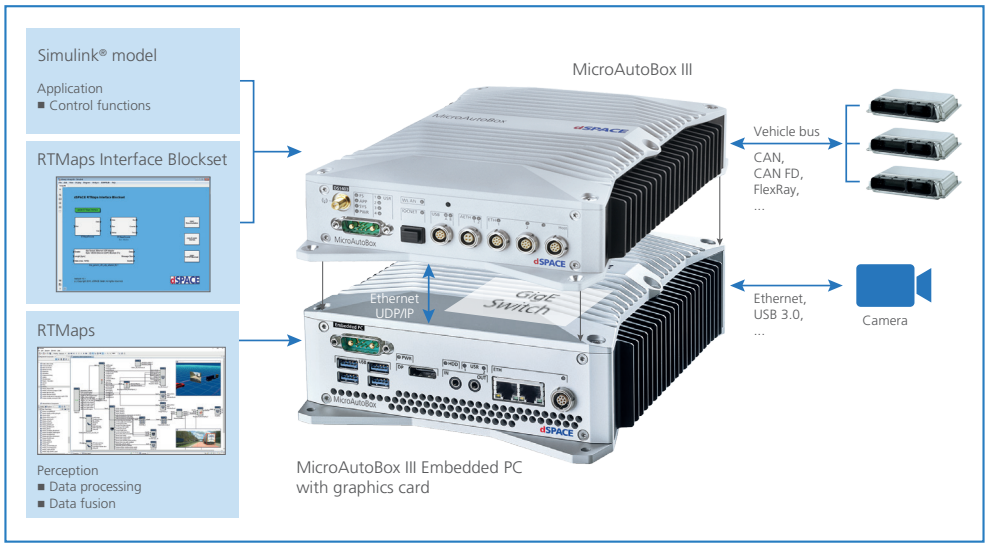
### Order Information

| Product | Order Number |
|---|---|
| RTMaps Interface Blockset | ■ RTMAPS_INTERFACE_SW |

### Relevant Software and Hardware

| Software | | Order Number |
|---|---|---|
| Optional | ■ RTMaps | ■ See relevant product information |
| | ■ VEOS | ■ See relevant product information |

| Hardware | | Order Number |
|---|---|---|
| Optional | ■ MicroAutoBox II | ■ See relevant product information |
| | ■ MicroAutoBox III | ■ See relevant product information |
| | ■ SCALEXIO | ■ See relevant product information |




Example setup for the development of perception and application algorithms.