Accelerating multi-sensor application development for Isobus-based agriculture machinery

Integrating the CAN-based Isobus (ISO 11783) communication into modern agriculture applications is a challenging task for engineering teams. Dspace (Germany) offers a development framework to address these challenges.

(Source: Adobe Stock)

As agricultural machinery progresses toward higher levels of automation, the development tool chains supporting these machines must evolve accordingly. Applications such as variable rate spraying, autonomous weeding, and precision harvesting depend heavily on vision sensors, such as cameras, and on Isobus protocols (ISO 11783) for communication with tractors and implements. Integrating high bandwidth camera inputs, machine control logic, and the structured communication flows defined by Isobus specifications remains a challenging task for engineering teams.

This article explores an approach to managing this complexity using Dspace tools – a GUI-based development framework in combination with an Isobus stack and simulation software – to support the development of advanced agricultural applications such as smart or spot spraying.

Challenges in multi-sensor off-highway systems

A precision spraying application can be considered a robotic system. Cameras generate continuous image streams, weed detection algorithms must run at low latency, and nozzle control must respond to changes in ground speed and implement motion. These perception pipelines must be time-synchronized with tractor-implement communication based on the Isobus standard. Engineers face some challenges:

1. Handling the high-bandwidth streams and timing constraints of vision pipelines.
2. Ensuring compatibility with Isobus conventions, including section control, prescription map handling, and implement reporting.

The result is a system where raw sensor data, algorithmic processing, and Isobus control logic must behave deterministically, while remaining flexible enough to be adapted as algorithms evolve.

Building a spot sprayer

To illustrate these integration challenges, consider an intelligent spot sprayer prototype. The system processes multi-camera feeds, detects crop or weed clusters, transforms detections into implement coordinates, and activates spray sections only where needed. At the same time, the implement acts as an Isobus client, providing the Task Controller with status information, volume application information, and work state feedback.

A key requirement in such setups is the coherent timing. Weed detections must correspond to the implement's actual position, even when the tractor's speed varies. The spraying decisions depend on relating image time stamps, tractor speed signals, and spatial offsets along the boom.

Before taking an implement to the field, engineers increasingly rely on simulation to validate interactions between perception and Isobus. A multi-layer simulation setup could include:

- A digital environment providing tractor motion and terrain conditions,
- 3D sensor simulation generating the camera feeds that drive perception, and
- Isobus simulation.

Dspace offers tools that help engineers accelerate development and validation. To demonstrate these capabilities, a concept demo was developed. ▷

Spot sprayer demo overview

The demo integrates a simulated tractor connected to a spot spraying implement. It includes two simulated cameras, a machine-learning-based weed detection algorithm, a nozzle activation controller (six sections), as well as Isobus Task Controller (TC) and Tractor ECU (TECU) clients (using OSB ISOsimulate). Isobus provides structured communication between tractor ECUs and implements.

The simulation is generated using the Dspace simulation tools ASM and Aurelion. Application development and execution is run on the Isobus RTMaps (real-time multisensor applications) middleware.

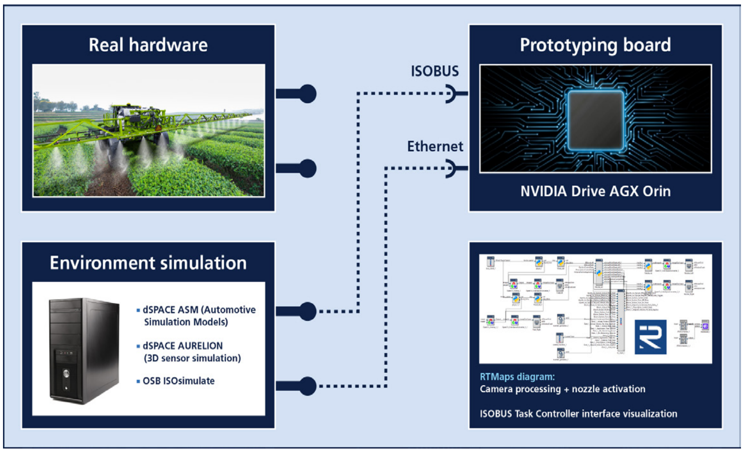

Figure 1: Demo application architecture (Source: Dspace, Adobe Stock)

The processing involves sensor simulation from ASM and Aurelion. The simulated raw image data is further processed on the RTMaps middleware. A deep learning-based weed detection algorithm running in RTMaps detects weeds in images and sends signals to the nozzle activation algorithm. The nozzle activation algorithm also receives signals from the Task Controller client, such as prescription maps and spraying methods, as well as tractor speed and GNSS (global navigation satellite system) data from the TECU client. Based on all inputs, it generates time-synchronized nozzle control commands and sends them back to the simulation to activate spraying.

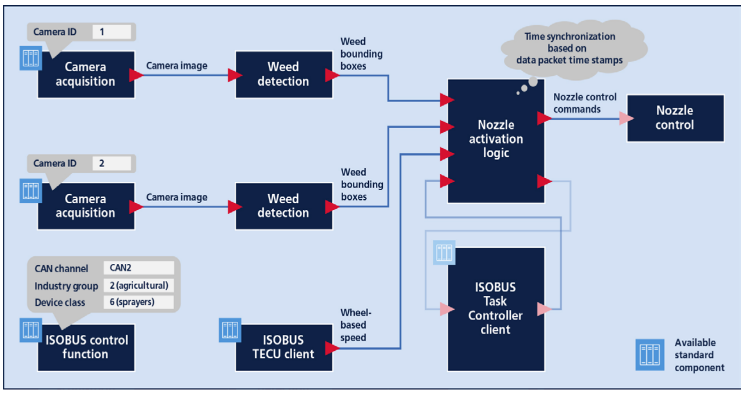

Figure 2: Graphical overview of application logic in RTMaps (Source: Dspace)

Visualization tools such as 3D scene viewers and telemetry dashboards complement this workflow. They help developers confirm that detections match the implement's physical geometry, that section command timing is correct, and that the Isobus TC receives accurate reporting information.

Bridging simulation, sensors, and communication

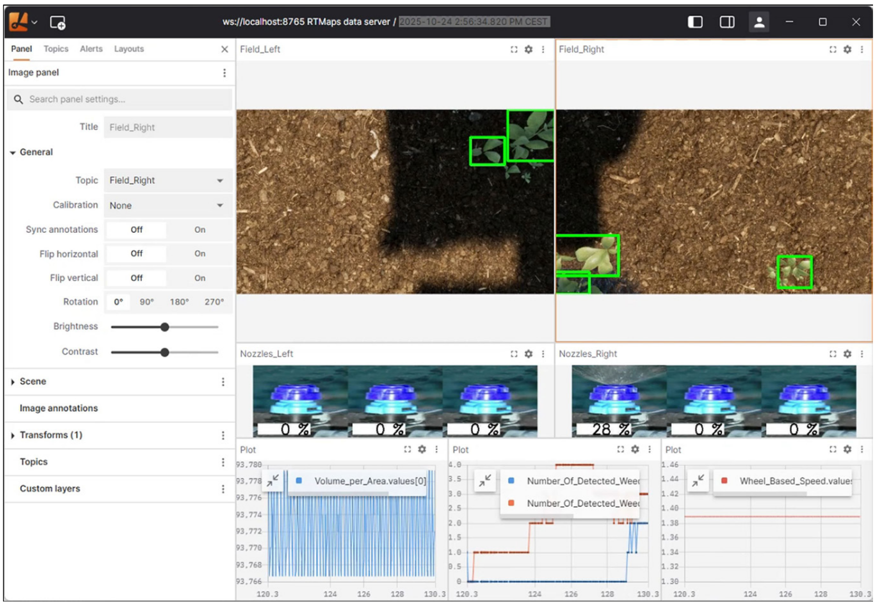

Figure 3: Visualization of weed detection and spray nozzles (Source: Dspace)

The core challenge in a system, such as the intelligent spot sprayer, is managing the sheer volume of high-bandwidth data (such as camera feeds) alongside deterministic control signals (such as CAN/Isobus). This is where dedicated middleware becomes critical.

In our architecture, we use the Real-Time Multi-sensor applications (RTMaps) middleware to handle this complexity. Rather than getting bogged down in low-level coding for sensor synchronization, engineers can utilize a dataflow diagram consisting of components, ports, and data connections. RTMaps also supports Isobus functionalities, allowing engineers to visually route parameters such as the setpoint volume per area application, actual section control states, and prescription control states.

The model-based design of RTMaps offers following benefits for application development and testing:

- Parallel execution: Components run in parallel and are triggered by events, ensuring high performance.
- Built-in synchronization: The system relies on built-in time synchronization mechanisms based on data packet time stamps, which are crucial when fusing camera data with vehicle speed and CAN telemetry.
- Complete lifecycle support: The platform supports the entire engineering workflow, allowing teams to record, replay, synchronize, integrate, debug, compute, and deploy.
- Collaborate workflow: Components and diagrams can be exchanged between colleagues and partners instead of ambiguous specs or code snippets.

A clearer path to production

By combining ready-to-use standard components for data acquisition and image processing with robust Isobus, camera, and tractor simulation, developers can validate their smart systems long before they hit the physical field. After algorithms have been validated in simulation and refined using recorded data, the final step is deployment to the actual implement controller. The RTMaps runtime allows dataflow diagrams to run on embedded hardware without rewriting the application logic. This helps teams maintain consistency between development, testing, and production environments – an important factor for complex, safety-critical systems.

Conclusion

As agricultural implements continue to integrate sensor driven automation with Isobus-based machine communication, middleware layers such as RTMaps offer an increasingly relevant architectural approach.

By managing sensor synchronization, execution timing, and communication routing, such platforms help engineers concentrate on algorithm development and system behavior rather than low-level plumbing. Whether developing smart sprayers, autonomous guidance systems, or variable-rate implements, this middleware-centric development workflow can provide a more stable and scalable foundation for next-generation agricultural automation. ◀

Authors

Peter Ngure
Dspace
pngure@dspace.de
www.dspace.com

Gautam Dobariya
Dspace
gdobariya@dspace.de
www.dspace.com