

# GPS WORLD

www.gpsworld.com

JANUARY 2004

Designing and Implementing Solutions With Global Positioning Technologies

**DESIGN:**  
DGPS on a Roll

**INTEGRATION:**  
Hosting an Ag API

**INNOVATION:**  
Antijam Protection

**SPECIAL SECTION**  
2004 Receiver Survey



## Two on One

An application programming interface hosts precision guidance controller on the GPS receiver and helps reduce soil compaction



# A Good Host

Programming Interface Takes Application Inside GPS

Kim Deimert, NovAtel Inc.

Rob Mailler, BEELINE Technologies Inc.

(C) BEELINE

BEELINE Technologies Inc. 2002

GPS- and inertial-guided automatic steering systems for use on large tractors contribute tremendous environmental benefits by reducing wheel traffic in the field and thus mitigating soil compaction, runoff, and diesel consumption as well as increasing crop yields. Existing systems with separate controllers were costly and complex, however, until one company developed an application programming interface to improve system manufacturability and reliability with software hosted on the GPS receiver platform.

In 1998, BEELINE Technologies began targeting the agriculture market with its automatic steering system developed for use on large tractors. The system provides highly accurate automated steering along parallel lines in a field, which are defined by establishing waypoints and then applying an implement swath width to offset each lap of the field, as shown in Figure 1.

The first and most obvious benefit of exact steering is the elimination of overlap, which reduces operating costs related to fuel and chemical use. However, less-obvious but much more significant benefits flow from strict control of vehicle traffic onto permanent wheel tracks. The field area affected by excessive wheel traffic ranges from 20 to 35 percent for each operation which, after six to ten operations for a typical two-year du-

ration, creates soil compaction in about 90 percent of the field. Using controlled traffic, on the other hand, eliminates soil compaction in the majority of the field, and, according to recent studies conducted in Australia (see "Further Reading" sidebar), can generate a 40-percent increase in yield due to improvement in crop root structure and an increase in the soil's ability to absorb and store moisture (see Figure 2). Improved rainfall storage also reduces runoff, thus decreasing erosion. Huge environmental benefits result as well, with diesel consumption reduced by up to 50 percent as a result of a reduction in draft, reduced implement width, and greater tractive efficiency on the permanent wheel tracks.

**Position and Attitude.** To achieve positioning within 2 centimeters of the target line 90 percent of the time, the automatic steering system uses data provided by a GPS receiver and an inertial measurement unit (IMU) to generate a position and attitude solution, as shown in Figure 3. The solution is then processed to generate steering commands that are provided to the steering controller through a controller area network (CAN) bus. Prompted by the commands, the controller activates valves to adjust the flow rate or pressure, and thus the steering.

To keep the position and attitude output accurate, the system-processes data provided by the IMU are processed using an error approximation, which is constantly updated by comparing each IMU solution estimate against data from the GPS receiver and then feeding the resultant error into a Kalman filter. In addition, the GPS receiver is aided by corrections from a satellite-based augmentation system (SBAS) satellites or a local base station combined with a wireless modem.

The design was well received but costly, and existing manufacturing arrangements constrained total volumes. In addition, due to its complexity, the need for highly skilled support staff meant it could only be sold in limited geographical areas that had existing service. BEELINE, however, had its sights on the global agriculture market, thanks in part to the emerging possibility of working with AGCO Corporation and its Challenger series of tractors, (see Figure 4,) which it re-

cently acquired from Caterpillar Inc. To successfully expand to the global market, the company had to reduce the cost and improve the manufacturability and reliability of the system. Obviously, to meet these requirements, the system had to be simplified.

BEELINE approached its suppliers to explore options for reducing the complexity of the system. They found that plans were already in the works to offer application hosting on the GPS receiver with an application programming interface (API), a set of predefined functions or building blocks for developing software to execute from and interface with the receiver platform.

Because the core of the product was the steering control software and the overall system design rather than the unique embedded controller, using the API to eliminate the controller hardware seemed to be the ideal method for simplifying the product while preserving the integrity of the company's technology.

### The API Concept

By allowing system integrators to host a custom application on a GPS receiver, the API eliminates the need for a dedicated hardware platform to support the application. This reduction in hardware enables lower overall product cost and increased manufacturability. Decreased reliance on typically troublesome components such as power circuitry and interconnects, along with the ability to complete upgrades in the field, result in lower maintenance costs. In addition, products developed with the API can often reach the market faster because of a shortened hardware-design cycle.

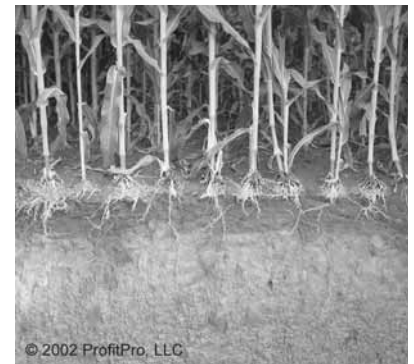
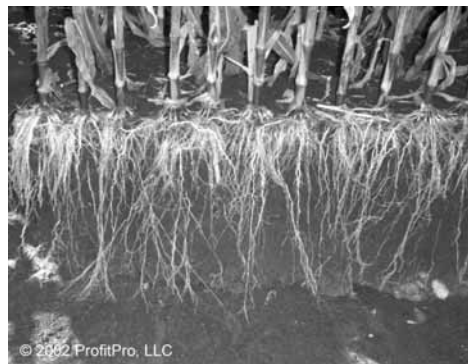
Developing a custom application to run on the receiver requires a standard C/C++ development environment and the API library of functions. These functions provide an interface to the receiver's hardware and firmware. Once built, the software is loaded into the receiver's memory using a supplied utility. The application is then executed using the receiver's standard command interface.

With the API, the application can interface with external devices through the receiver's serial ports. The application's main interface to the receiver is through virtual ports, which offer the same communication



BEELINE Technologies Inc. 2002

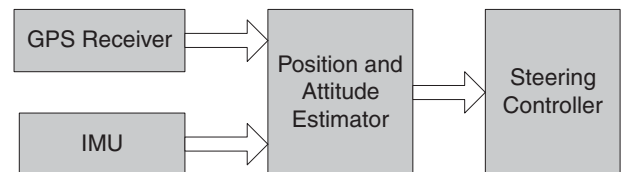
▲ FIGURE 1 A tractor using an automatic steering system for highly accurate tilling.



▲ FIGURE 2 A healthy, massive corn root system (left) that can ingest ample amounts of nutrients and moisture is the key to high yields. Corn roots in compacted soil (right) grow horizontally and can't establish the deep mass and strength necessary for excellent plant stability and high yields.

functionality as physical ports but exist entirely in software. This setup replicates the interface to an external receiver and reduces the time necessary to port the software. Advanced functionality such as task prioritization and message queuing is included to ensure the receiver can support complex applications. The API also provides mutexes, designed to prevent multiple processing tasks from accessing a shared resource concurrently. Semaphores, used for signaling between tasks, are also included.

Early on, this application hosting functionality emerged as the key to simplifying the BEELINE system. Using the receiver to support the control software and providing an integrated tractor control interface



▲ FIGURE 3 The components of an automatic steering system.

would allow the development team to eliminate a large section of the hardware from the design, as shown in Figure 5. The API appeared to be the optimal solution for meeting the company's requirements for manufacturability and reliability within AGCO's schedule.

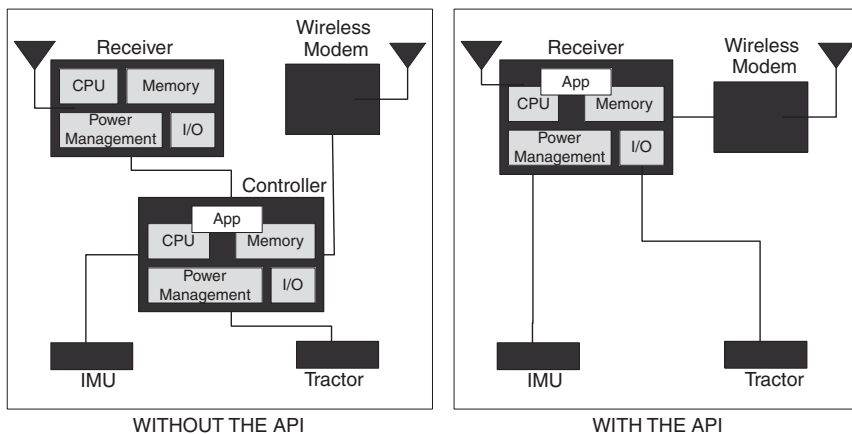
### System Development

BEELINE's new product design featured four methods of steering control to support multiple tractor configurations. As a result,



BEELINE Technologies Inc. © 2002

▲ **FIGURE 4** ACGO Challenger setting a world record for the number of acres planted in a 24-hour period, drilling seed and fertilizer in 1413.19 acres in a single day in the Sinelnikovo region of Dnipropetrovsk, Ukraine, on May 29, 2003.



▲ **FIGURE 5** The steering system without the API (left) and with the API (right). Using the API, a large section of hardware can be removed from the system.

it was necessary that the API, along with the receiver, provided a CAN 2.0b bus interface and control of three low-voltage transistor-transistor logic (LV-TTL) general-purpose input/output (GPIO) signals. The GPIO signals are used as a set of two outputs for steering right or left and a single input for aborting steering control. API support for the receiver's digital pulse-width modulator output was also required for an alternate method of steering in which the length of time the signal is pulsed high, translates proportionally to the pressure in the valves.

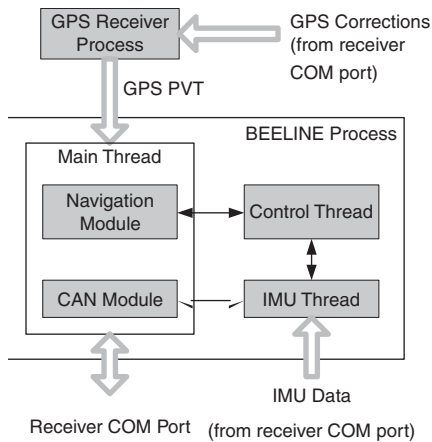
**Designing the Prototype.** Within months of initiating the project, the developers delivered a prototype of the API, which was generated in parallel with BEELINE's work. BEELINE's developers began the process of porting their existing control software to the receiver platform and adding the new functionality required for the product. An immediate challenge was the non-availability of the debugging facilities that are essential for any software development project. To that end, the company built an offline simulation environment on a PC to replicate the API, allowing developers to

run the BEELINE application on a PC target and access more-extensive debugging facilities. The software was designed to compile from the same code base to target either the receiver or the PC simulation. Although this method worked well, eventually differences in behavior between the simulation API and the receiver platform — such as difference in CPU availability and execution time for floating-point operations — made the debugging process more difficult.

Given that BEELINE's steering technology included complex, CPU-intensive algorithms, the effective use of CPU time was a key requirement of the application. Specifically, reliable, low-latency access to the receiver's serial ports between 10 and 25 milliseconds was essential to ensure the data from the IMU could be accurately sampled. The API prototype was designed with multiple task priority levels, all below the receiver's main tasks for acquisition, tracking, and processing of the GPS signals. However, initial testing showed that BEELINE's software was unable to consistently read from the serial port with minimum latency because the higher-priority GPS processing prevented the company's tasks from running at the necessary time.

**Thread Priorities.** With the API still in development, a priority level above the receiver's main processing tasks was added for such situations, with the understanding that use of this high-priority task level required care. If a complex, lengthy task was executed at the highest priority, the receiver performance could be compromised due to lack of resources. As a result, BEELINE placed considerable effort toward the careful management of thread priorities to deliver the required performance and serial port access without sacrificing the integrity of the receiver's position data.

With multiple real-time threads (see **Figure 6**) running and competing for processing power, re-creating situations for debugging purposes also became difficult. Because the ease of debugging is crucial to any development effort, functionality was added to provide output of the registers and the memory location from where the application was executing when an error occurred. Finally, enhanced memory protection was



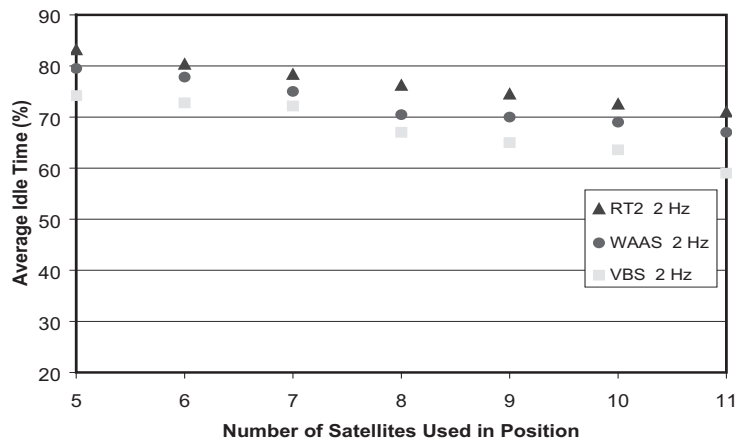
▲ **FIGURE 6** The BEELINE steering system consists of multiple threads, each with a specific purpose and interface to the receiver, tractor, or IMU.

put in place. Defining boundaries for the memory space available to the application eliminated the possibility that the memory used by the receiver could be corrupted.

### Integration and Testing

With the API and the control software functionally complete, the team executed initial tests with a standard off-the-shelf product consisting of an OEM GPS receiver in a metal enclosure with three RS-232 serial ports, and a prototype tractor that became available just days before the trial. Because up to that point BEELINE had developed from documentation, these initial tests proved to AGCO that the concept was valid. Later, with prototype receiver hardware that integrated an OEM GPS engine, an SBAS L-band receiver to provide corrections, a CAN bus interface, and multiple GPIOs, the developers completed more-extensive testing to ensure sufficient receiver memory and processor capacities were available to support the application while maintaining position integrity.

Testing also confirmed the application had reliable access to a serial port capable of the high data rates needed for a responsive system. Given the expected increase to quadruple the previous product's volume, which would prohibit the type of direct support previously offered, BEELINE required at least a one order of magnitude improvement in product reliability. As a result, the receiver hardware was put through severe independent environmental testing before it was approved.



▲ **FIGURE 7** The amount of CPU time available to the application is affected by the number of satellites used to calculate the position as well as the type of correction data in use. (RT2 = 2-centimeter RTK corrections; WAAS = WAAS corrections; VBS = OmniSTAR VBS service corrections).

**Stabilizing System Design.** The first BEELINE product deployed using the API was a submeter, entry-level product that did not require the highest level of system accuracy on the ground. With this product, BEELINE was able to demonstrate that the basic functionality had been achieved. Each of the products in the family was designed with the same GPS receiver hardware and control software and differentiated only by the correction data used and the positioning algorithm accuracy. Therefore, once the system design stabilized for the lower-accuracy product, the company was able to immediately begin working to deploy its midrange, decimeter product.

This step exposed a range of optimization issues that the BEELINE team worked through and extensively tested. At the same time, NovAtel continued to improve the API and firmware, which set in motion additional rounds of field testing by BEELINE. Continuous 24-hour testing on tractors carrying out a commercial operation was an effective way to capture a range of GPS constellation conditions and a range of operational procedures typically completed by the end users.

The biggest challenge was delivering the highest-accuracy product based on the receiver's 2-centimeter RTK positioning. With the extensive CPU time required for the product's complex steering algorithms, considerable testing of the system in field conditions helped discover rare CPU starvation issues that were not easy to catch on the bench. As Figure 7 shows, with all signals in

view being tracked, the changing nature of the GPS constellation — and, therefore, in the configurations of specific satellite signals being used in successive positioning calculations — delivered variability in the available CPU time, and extensive testing was required to explore the depth and breadth of the variation.

To match the same performance as that of the previous generation of products under all conditions, BEELINE developed several techniques, including optimized trigonometric routines that ran more than ten times faster than standard library functions, leaving more CPU time available for other operations. The company developed a sophisticated satellite selection module that reduced the system load associated with processing the data by removing whatever satellite tracking did not contribute to the solution.

To facilitate the optimization efforts, some of the internal rates involved in the tracking loops of the receiver firmware were altered as well. With the rates required by the application aligned with those required by the receiver, a large performance boost in available CPU time resulted. This final breakthrough demonstrated that the API approach could deliver an automatic steering system without performance constraints.

### Results

In the end, while the process of moving the highly complex BEELINE software to a smaller CPU required a number of optimizations, as is typically necessary for in-

tegrating a highly embedded system, the system performance was enhanced on many levels. And though implementing the application at the same time the API was developed made the path more difficult, it did allow for the opportunity to influence the design of the API. From a cost standpoint, by adding support for the API and the physical control interface to the receiver, the company's per-unit material cost was reduced by \$500 USD when the elimination of components and the increase to the cost of the receiver and mounting hardware are taken into account.

However, measuring the other advantages of the API over the previous design is not straightforward because the benefits were not limited to a simple product replacement. In truth, accessing the API allowed BEELINE to transform its business. More than ten times the volumes can be manufactured than could be reasonably manufactured with the previous-generation product. Manufacturing arrangements have been simplified by significantly reducing the number of hardware suppliers, which has translated into a business model better suited

### Further Reading

For more about the effects of wheel traffic in the field, see

*"On Track" to Sustainable Cropping Systems for Australia*, by J.N. Tullberg, D.F. Yule, D. McGarry, proceedings of ISTRO 16: Soil Management for Sustainability, p. 1271, 2003.

*Yield and Protein Variation within a Controlled Traffic System*, by T. Jensen, G. Powell, T. Neale, proceedings of the 10th Australian Agronomy Conference, Australian Society of Agronomy, Concurrent Session 1, Jan. 29, 2001; <http://www.regional.org.au/au/asa/2001/index.htm>

"Wheel Traffic Effects on Tillage Draught," by J.N. Tullberg, *Journal of Agricultural Engineering Research*, Vol. 75, No. 4, pp. 375-382, April 2000.

for managing a global distribution system. In addition, the receiver hardware delivered more than a one order of magnitude increase in the reliability of the system, and its use eliminated the cost and time of regulatory testing by BEELINE. All these improvements were essential for accessing the global market, and most would not have been possible without the API. 🌐

KIM DEIMERT leads the OEM Software group at NovAtel Inc., a precise positioning technology company, and was the core developer of the API library and supported BEELINE's porting to the receiver platform.

ROB MAILLER has a background in control systems and mathematics and is the chief technologist at BEELINE Technologies Inc., which he founded in 1994. He was involved in all levels of the Steering Assist system design and implementation.

### Manufacturers

BEELINE's *Steering Assist* system includes a custom GPS receiver from NovAtel (Calgary, Alberta, Canada) that combines an OEM4-G2 GPS engine with RT-2 functionality, an L-band OmniSTAR (Houston, Texas) satellite receiver, and a CAN bus interface. The L-band receiver uses OmniSTAR's high-performance or virtual base station service. The system also incorporates a Crossbow (San Jose, California) inertial measurement unit and a wireless modem by Freewave (Boulder, Colorado).

© Reprinted from GPS WORLD, January 2004 AN ADVANSTAR PUBLICATION Printed in U.S.A.

Copyright Notice Copyright by Advanstar Communications Inc. Advanstar Communications Inc. retains all rights to this article. This article may only be viewed or printed (1) for personal use. User may not actively save any text or graphics/photos to local hard drives or duplicate this article in whole or in part, in any medium. Advanstar Communications Inc. home page is located at <http://www.advanstar.com>.

