MINERAL EXPLORATION

Overview:

ASG has supported projects requiring knowledge of gravity measuring techniques used in mineral and oil exploration. ASG has been involved in all software aspects of these projects: real-time instrument monitoring, calibration and stabilization software, data communications, data analysis, operator interfaces, data collection and archiving, failsafe operation, low-level interrupt services, and critical timing algorithms. These very extensive projects have involved several staff members with diverse backgrounds, programming requirements that included knowledge of Assembler, Pascal, C, C++ and Visual Basic, and low-level understanding of three different operating systems, numerous processing platforms and peripheral equipment, plus real world experience with instrumentation and industrial environments.

Project Description:

Applied Sciences Group participated in the development of a series of gravity gradiometric measurement systems. These gradiometers are sensitive instruments used to detect changes in the gravity field over an area of land or water, and as such are useful for oil, water and mineral exploration. Each system comprises three elements:

- A gyro-stabilized platform containing an array of accelerometers, which senses changes to the gravitational environment
- A real-time control program that maintains platform alignment and counters the effects of the Earth's motion through space and the transport vehicle's motion across the Earth
- A program to process the sensor inputs in real-time and to display these results on a console.

Applied Sciences Group has been involved in all control and MMI aspects of the gradiometer program development. ASG assisted in the design of the platform calibration, platform stabilization and gravity instrument control algorithms; and is principally responsible for the design and implementation of the real-time task manager and control program that drives the entire system.

Approach:

Sensor data is shuttled from the control processor to an analysis processor via Ethernet for subsequent processing (also in real time). Here, ASG developed the process algorithms and the man-machine interface, used for both system control and to visually observe interesting gravitational events.

Results:

The gradiometer system was developed using a common design and programming methodology, which was adapted to support each customer's specific needs. The maximization of common software components (and minimization of components unique to a customer) was given a high priority in order to keep costs down. Additionally, several different programming languages - C, Pascal, Assembler and PVWave - were selected for different subsystems, based on the overriding real-time performance issues.

POWER GENERATION SIMULATION SOFTWARE

Overview:

The control systems for typical power generation facilities are complex and revisions to the control code are generally not available for online testing prior to use. ASG has been providing support for the development of simulation software that augments the testing process, to increase confidence that the software will not fail. Software is also being developed to support remote monitoring and display of key

power generation parameters. The software is written mainly in C++, VB and VBA, with hooks into Intellution IFix SCADA software and Honeywell DCS.

Aircraft Communications

Overview:

Aircraft flying in close formation require constant communications between them to eliminate the possibility of midair collisions. Military aircraft in close formation do not always have the luxury of visual ranging, as some of their operations require nighttime flights or flights in bad weather. ASG was part of the team that is developing the next generation of military "station keeping" software. The software is written in Ada, C++ and Visual Basic and relies on shipboard radar for detection of aircraft and communication between aircraft. Most of the software that ASG developed was dedicated to maintaining the reliable communications channel, and simulation.

Heart Monitoring System

Overview:

A medical device developer requested help in developing the software for a new type of heart monitoring device. Software development is guided in part by 21 CFR Part 820, which provides the framework by which the software is designed, written, tested, modified and managed.

The software comprises a control system for the electromechanical portion of the device, a set of data collection and operator interface functions to display EKG, heartbeat, blood pressure and other parameters, and a diagnostic function containing the signal analysis algorithms needed to diagnose specific heart conditions. The software is written in Visual Basic and C++, using an Access database for storage of patient data.

Music and Sound Arrangement Software

Overview:

ASG is involved in two separate projects in the music industry; both are related in that they require knowledge of "WAV" file formats and software to visualize and manipulate the data. One project - written mainly in C++ - involves the development of digital compression techniques to fit the end product into the smallest possible footprint. This has the added benefit of maximizing the data throughput through any communications channel. The second project - written in Visual Basic - gives piano arrangers the ability to display and modify WAV files, create an extremely unique and proprietary musical format, and in the end produce a popular marketable product.

Network Management and Remote Monitoring

Overview:

In some cases, companies with products, offices or production facilities at distributed sites have been hard-pressed to centralize communications, monitoring and control to/from these sites. In particular, the manufacturing industry and many financial markets rely on centralized communication to determine process efficiency, downtime and collection of the data that is needed to make quick decisions.

ASG has been involved in the development of both network management and remote monitoring software for custom applications. The lottery and ATM industries, for example, require status criteria from every terminal within a network of terminals, with real-time delivery to a central site. In some cases, statistical process control techniques are used to analyze "hot spots" or other anomalies that could lead to terminal or network failure. ASG has supported the efforts of several customers to develop software for network

configuration, terminal configuration, remote monitoring and remote download. Many of these efforts used proprietary techniques and were written from the ground up using C++, C, Visual Basic, Ada and Assembler; others involved off-the-shelf software and relied on existing formats such as SNMP for communications.

Stock Car Racing Support

Overview:

Vehicle telemetry and knowledge of track conditions are critical to professional race car driving, where racing teams spend millions of dollars each year and where the difference between winning and losing is often measured in fractions of a car length. ASG has been involved with a stock car racing team, developing custom software to help give them an edge on the competition. This software deals with a combination of vehicle parametric information and track data to provide vehicle performance estimates used throughout the race. The software is written in Visual Basic.

On the table is a proposal that utilizes ASG's wireless expertise to provide a track-wide communications network, for the "closed circuit" broadcast of a number of track and vehicle parameters used by all racing teams throughout the race. During a race this data will also be made available to racing fans via the Internet. It is to be written in Java and C++.

Wafer Mapping Software Interface

Overview:

A local electronics manufacturer asked ASG to design, develop, and implement a Hughes 3500 System Interface. This project involved the development of a user interface to perform the function of transferring wafer data from a 3.5" diskette directly into a Hughes 3500 System for quality control purposes. Specifically, the software package was written using Visual Basic for the operator interface and Visual C++ to generate the required DLL functions. This interface solution enabled the down-loading of wafer "availability maps" from manufacturer supplied diskettes (3.5" format) into the Hughes 3500 System. Object availability status was updated on a continual basis and saved to a temporary disk file, which could be used for backup or power failure events. Upon the introduction of a new wafer, the operator inputs the identifying serial number and the associated availability map is transmitted to the Hughes 3500 System.

Unidrive Modbus+ Interface

Overview:

The Unidrive Modbus+ Interface involved the software engineering, programming and support effort required to design and develop a fully functional Modbus+ interface for a "family" of OEM control drive units. The executable program resides on the customer's OEM co-processor and uses the application's programming interface (developed by the customer).