Heterogeneous Device Networking

by William Foard, Foard Systems Design, Inc.

FOARD SYSTEMS DESIGN HAS DEVELOPED A NOVEL APPROACH TO NETWORKING SENSOR AND CONTROL DEVICES IN DISTRIBUTED SYSTEMS, ALONG WITH CONTROLLERS, DEVICES, AND SYSTEM CONFIGURATIONS BASED ON THIS NEW TECHNOLOGY.

Foard Systems Design has developed a novel approach to networking sensor and control devices in distributed systems, along with controllers, devices, and system configurations based on this new technology. The Atmel AVR family of microcontrollers has played a central role in these designs, since AVR features are ideally suited to the diverse demands of the application.

FGI Distributed Sensor and Control Network Technology

The FGI networking technology provides an easy-to-use, effective, high reliability, low power consumption, and low cost means for interconnecting a heterogeneous mix of remote sensor and control devices. Configurations with thousands of devices can be practically implemented.

At its core, the FGI networking approach uses a modified 4-20 milliamp signaling technique, where devices on the network are individually enabled to transmit on a common 4-20 ma signal line. Control signals superimposed on the 4-20 ma line, plus a separate data transmit line are used to select device activation states and to send control data to the devices.

Simple devices are implemented with direct connection to the FGI networking logic. For networking 4-20 ma sensor types, their output current is switched directly onto the FGI 4-20 ma signal line, and digitized at the controller. This maximizes network performance, since no packet protocol is needed for device selection or data measurement. Similarly, simple digital devices such as switch sensors are implementing by directly sending high-low current levels on the signal line. In combination with control data sent to the device, multiple input and output devices can be implemented to directly support such features as latched outputs, frequency sensing, rotary encoder inputs, and direct-scan LED module outputs. This provides remote access to temperature sensors, digital potentiometers, video switch matrices, and many other commonly available components. More complex functionality, such as motor control, benefits from incorporating device level microprocessors. The FGI interface provides excellent microprocessor support. Devices can implement master-slave communications, keeping the microprocessor size, cost, and power requirements to a minimum. Alternatively, devices can implement peer-to-peer communications for more immediate performance, with the FGI networking chip providing most of the packet collision avoidance functionality. Master-slave, peer-to-peer, and non-microprocessor analog and digital devices can all be intermixed on the same network.

Additional support for AVR microprocessors provided by the FGI networking design includes a means to remotely reprogram the device microprocessors via the network through the SPI interface, without the need for or the overhead costs of any self-programming support hardware or software in the device. Single point remote programming can deliver feature enhancements, bug fixes, support for new attached hardware, customization features, and temporary diagnostic software.

The FGI interface logic also provides plug-and-play style self-configuration and built-in network self-diagnostic capabilities. Serial EEPROM support is provided in non-micro devices to store additional device information such as calibration data. Multi-featured combination devices are also supported.

FGI network devices are connected to an FGI network controller through a branched daisy chain topology, simplifying initial installation and future system expansion. Power, ground, and two signal lines are used to interconnect devices.

With many device types requiring as little as 3 milliwatts per device, overall system power requirements are held to a minimum, much simplifying large scale system implementation. The miserly power requirements result in smaller network power supply and backup battery needs, with resulting smaller space, heat dissipation, and weight benefits. Importantly, the low current requirements allow much smaller wires and connectors to be used for interconnecting the remote devices, with resistive losses held to a minimum. For devices such as temperature sensors, the low power also minimizes inaccuracies introduced by self-heating.

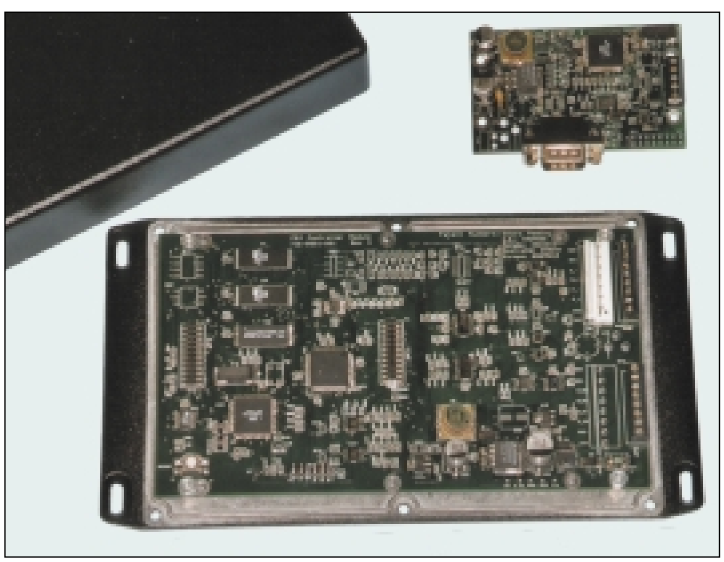
The FGI network controller communicates with a host computer or other controller via RS-232 or RS-485, with additional interface options planned. Support software is provided, including serial-to-TCP/IP bridge software.

Atmel AVR Feature Use:

AVR Based Network Controllers

The FGI network controller is based on an ATmega128. This full-featured processor minimizes support components needed to implement the controller, reducing size, increasing reliability, improving manufacturability, and minimizing cost.

The FGI network uses 4-20 ma signaling from the remote devices. The AVR's internal 10-bit A/D is used to digitize analog 4-20 ma measurements, maxi-



FGI Controllers

mizing the measurement rate performance of the network and eliminating the need for and, cost of incorporating A/D and data packetization capabilities at each device. The AVR's internal comparator is used to convert high-low current measurements into bit levels for digital signaling and communication fault-detection.

The AVR's run-time remote self-programming feature is used to provide field software updates. With its secure boot sector, password access control is implemented, guarding against unauthorized access to the control system.

The controllers also provide slave port access via the AVR's SPI interface, so external access is available for full diagnostics and boot sector reprogramming.

The FGI controller uses the AVR's internal 128K FLASH for program space. An external 128K SRAM fills out 32K of fixed SRAM in the lower half of the AVR's data address space, with an additional 100K SRAM accessible via paging in the upper half. An Atmel small-sector FLASH provides 1MB of device configuration data space, also via paging.

A smaller FGI controller design operates with the ATmega128's self-contained memory only, minimizing the implementation cost for small dedicated applications.

One of the AVR's dual serial ports provides host access via RS-232, RS-485, FGI slave, and SPI programming access, with the other dedicated to FGI network device communications.

The AVR's internal 4K EEPROM is used for controller configuration parameters, and storing manufacturing and test history.

At 3.3V the AVR provides processing power well suited to current needs. However, having room to grow is appreciated, with a 5V AVR option available to double the controller's horsepower while maintaining software compatibility.

Future support for direct Ethernet TCP/IP support is planned, based on the AVR's TCP/IP development kit available from Atmel. PC-104 and modem options are also planned.

AVR Based Network Devices:

The AVR family is ideal for implementing microprocessor based FGI network devices. The wide range and variety of AVR sizes and features available allow for tailoring the features, size, power consumption, and cost to the needs of a widest possible range of devices, while retaining software language compatibility across the controller and all the devices.

Minimum size, cost, and complexity microprocessor based devices can be implemented using the tiny AVR devices. An 8-pin micro with internal RC oscillator and no external micro support components can be used for implementing a master-slave micro device, communicating with the controller via bit-banging or the SPI interface. Mid-sized and larger AVR devices can be used to implement FGI devices with full peer-to-peer communication capabilities.

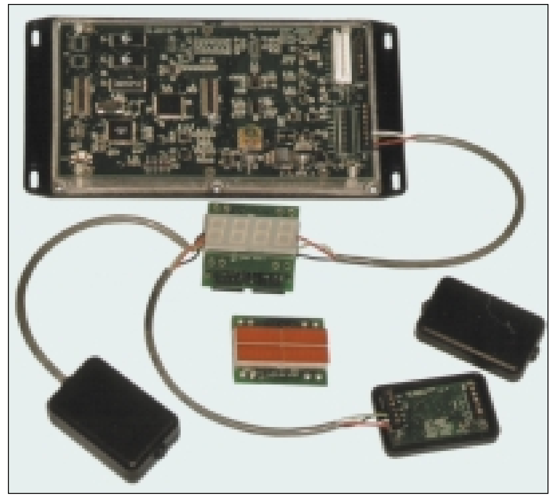
The AVR's SPI programmability via the FGI network provides capabilities for system-wide remote updates, build-in-test, manufacturing help, and fail-safe operation.

The FGI network's philosophy of implementing minimum power devices is supported by the AVR's power-saving idle and sleep modes. A master-slave micro device can spend most of it's time powered down, waking only when accessed by the controller and as needed to satisfy other device-specific needs.

FGI devices also use the AVR's EEPROM space to store additional device specific information, such as manufacturing lot, calibration, and device configuration parameters.

Example System Configurations and Applications

The FGI Temperature and Humidity Monitoring System provides periodic monitoring of temperature, humidity, external 4-20 ma sensor measurements, and external switch states.



FGI Systems

The FGI 4-20 ma Sensor Networking System provides a convenient means for interconnecting 4-20 ma sensors, rather than using home-run connections for each sensor. Single and multiple input devices can be provided, and control output devices can be supported on the same network.

An upcoming FGI Modular Custom Control Panel System provides a modular approach to implementing displays and control panels of varying complexity. LED driver devices can be combined with a variety of display options. Switch, keypad, and rotary encoder input devices provide for user interaction.

We are also using this technology to design an automated test system for functional testing of manufactured electronics products.

Example end-user applications include manufacturing control systems, laboratory process monitoring and control systems, refrigerator/freezer monitoring systems, and building automation systems.

Future Vision

FSD is presently giving priority to supporting industrial and commercial measurement and control system applications. In support of longer term goals, FSD is also developing application level software for using the FGI networking technology for home and commercial building automation. The FGI networking technology can provide substantially more features and functionality at lower installed cost than competing technologies, and the volume potential of these markets can drive implementation costs way down, also benefiting our industrial automation and commercial users.

To reach these larger markets, we would like to eventually partner with and/or license our technologies to semiconductor and 3rd party device suppliers, and promote the FGI networking technology as a standard OEM product device networking approach.

❑