Minimum capabilities to fit in to existing Control System environment:

* Means to send commands to and monitor state of device (ModBus and CA currently used)
* Means to interact with the device while it is running
* Continuously updated event log backed up to a server (so we have some information to work from when problems DO occur)
* Ability to update the firmware without physically accessing the device
* Ability to automatically discover and configure a device and to monitor it (to detect conflicts with other devices)
* Support for Ethernet standards:
  + UDP: For automatic monitoring/discovery/config and support of the CA protocol
  + TCP: For CA and/or ModBus protocols, login shells, remote file services, etc
  + ARP: Required for UDP and TCP to function

Key aspects/capabilities of DEBROS-based controllers:

* Industry-Standard development environment (adheres to Unix/Linux standards, including BSD-compatible socket library)
* Priority-drive, pre-emptive multitasking model: Compared to a co-operative multiprocessing model, pre-emptive multitasking results in much less code, and code that is much easier to write and maintain. It also translates much easier to a parallel processing model.
* Interactive diagnostics (multiple login shells and many commands to monitor and gather information on the state of the device)
* Continuously backed-up event log
* Swapped in devices automatically discovered, configured, and loaded with the appropriate device calibration data and persistent settings
* Ability to update persistent configuration values while running
* Device-specific configuration data managed separate from the control system