Building The Industrial Cloud

Moxa America

There is a pressing need amongst industrial automation suppliers for embedded computing platforms that are optimized for machine-to-machine (M2M) communications. 2013 demonstrated that the Internet of Things is poised to soon become a material reality, making great possibilities for industrial providers. Yet until the lack of specialized platforms is overcome, the building of IoT clouds for industries like traffic management or the smart grid will continue to be delayed. The fundamental challenge in designing these systems involves negotiating the divide between IT and IA technologies. Industrial automation protocols are fundamental to the proper functioning of edge devices that make up the mass of mass deployments; fieldbus, Modbus, input/output configuration, serial interfaces, and the gateways that link all of these are common enough problems for industrial automation engineers, but for most IT engineers the entire field appears esoteric and mysterious. On the other hand, securing networks with firewalls and VPNs, protecting against dropped packets or node failures, and the multifarious problems introduced by wireless communications are all things typical for IT, but which IA engineers would rather avoid.

The biggest challenge today when building an industrial M2M network is its massively distributed nature: there is very little value gained by a gradual transition that takes years. IoT deployments must be rapid enough to sustain investment value, but the huge number and scope of the cloud of devices employed makes speedily completing the full installation nearly impossible. Specialized technicians are required for every individual station, and these men and women must be both competent electricians as well as familiar enough with the network design to juggle the details of multiple protocols, interfaces, and communications media. Unfortunately, few IT professionals have ever worked with the protocols and interfaces that are most common to industrial automation networks, and setting up input/output stations, or configuring sensors, are things which IT people simply have no experience in.

Thus, at the outset, there are two important reasons why effective software automation at the connectivity layer is critical: first, to facilitate the deployment of a cloud of devices that may ultimately include hundreds of thousands of nodes, and second, to flatten the learning curve as much as possible for the men and women who will be installing these devices. Software tools that transparently automate the rollout of industrial automation devices and simplify overall network deployments are, therefore, a foremost consideration for anyone who will be managing the deployment of an IoT network.

Industrial-Grade Stresses, Enterprise-Class Challenges

Yet another way in which IoT networks significantly differ from consumer networks are their strict availability and reliability requirements. Industrial M2M systems for

intelligent transportation systems or the smart grid will, of necessity, operate 24/7, 365 days of the year. At literally any moment the network must be able to call upon remote stations at the network's edge and command them to make adjustments, return data, or perform maintenance checks. Obviously, devices which are not capable of reliably maintaining network connectivity for years on end will not be very valuable to network administrators. Similarly, all devices along the network must be able to deliver key information for preventive maintenance, and to respond to a wide variety of common network challenges such as failed nodes, network congestion, and wireless re-association. For the most reliable performance, network redundancy, automated connectivity checks, preventive maintenance routines, and effective maintenance, monitoring, and control protocols must be integrated as deeply into the hardware level as possible.

Finally, there is the consideration of the related problems of data integrity and network security. Authorization, access, and accounting controls are as imperative for an M2M network as they are for any IT network. Authorization and access are easily understood: illicit access to a massively distributed industrial network by a hostile party has clear potential for disastrously lethal consequences. For this reason, IoT networks (especially those for traffic systems, the smart grid, or other power applications) must support the strongest possible encryption and access controls. Similarly, accounting controls over the entire network are important not only for the monitoring and management of the network itself, but also to aid in preventive maintenance, as well as to perform forensic analysis on suspected intrusions, or other security breaches.

Taken together, these imperatives amount to a lot of work that, until recently, had simply not yet reached a stage where machine-to-machine communication networks could be considered viable. Now, however, that has changed, and the IT/IA convergence of recent years has arrived at a point where integrating these two technological realms in a secure, reliable, cost-effective manner has become a relatively easily achievable reality.

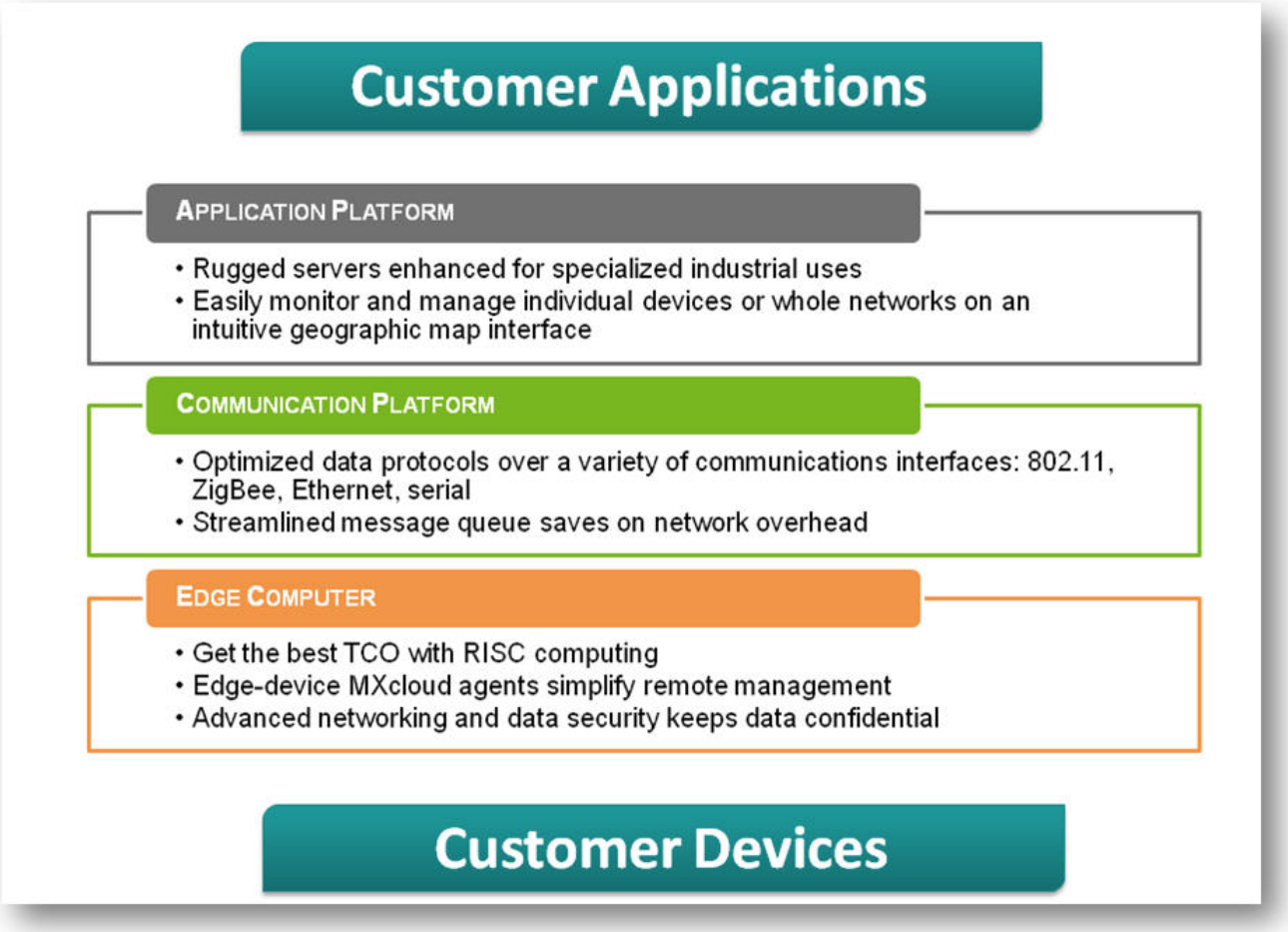
Converging Solutions for Converging Technologies

The overall technical challenge that must be addressed when building an industrial IoT network may be broken down into three key aspects:

- □ network topology (and how that relates to device deployment, and engineering);
- □ the deployment, setup, and maintenance of network nodes and edge stations; and
- □ overall monitoring and control.

First, let us consider how an M2M network can be envisaged in terms of its topology, and how the technological implications of that structure affect the network's design and implementation. At first glance, it might be tempting to simply divide the network into two layers of two dimensions: network/process, edge/core. That, however, would neglect some important opportunities for automation and

optimization. To begin with, the physical devices of the network are best broken up into three concentric layers, rather than two: the core, the connectivity layer, and then the terminal edge stations, where nearly all of the remote process data and events will be generated. Edge devices will necessarily include sensors, automatic metering infrastructure, embedded computers for control and monitoring, and gateways to bind all of these devices together, to allow effective communications between the various parts. The question then becomes: how can automation and effective device engineering speed up and simplify M2M deployments, monitoring, and management?



Flexibility that Simplifies Development and Adaptation

To guarantee that the network remains as customizable and flexible as possible, open platforms should be utilized wherever they are prudent. Linux/GNU and other open source solutions provide an excellent platform for IoT integration, and may reliably power both RISC and x86 platforms. These proven software solutions offer strong security (for both data integrity and AAA protocols) while providing a wide-open system that allows customization, optimization, and feature development on any subsystem process, no matter how low- or high-level it may be. Linux/GNU systems also offer two additional advantages: strong security in the form of packet filtering, firewalls, VPNs (and the strongest RSA encryption available), along with the important benefit that end users can escape the danger of proprietary lock-in,

whereby a device may become useless should the manufacturer one day disappear, or decide to cease support for that particular line of hardware. Thus, system integrators and end users alike benefit powerfully when using open source/free software solutions like Debian.

Software optimizations are not, however, the only consideration. The physical devices that make up the IoT network must also be specifically engineered for customizability, security, reliability, and deployment flexibility. For embedded computers, features like a modular design are a critical feature that will allow end-users to adapt devices to specific roles within the network, or even to repurpose a device that is being used in an obsolete role. A wide variety of communications modules must be available, as well: ZigBee, Ethernet/IP, 802.11, cellular, and fiber must

Five Principles to Guide the Engineering of an IoT Platform

Taking all of these observations into account, a clear vision emerges of what kinds of embedded computing platforms should be sought out when preparing to build an IoT solution.

1. IoT networking devices should conform to the strictest standards of flexibility, reliability, and security, starting with the physical hardware and then moving on up through every networking layer, right into user-space.

2. Software optimizations that automate configuration, setup, and the overall deployment of embedded computers and other edge devices are critical components of an effective IoT architecture.

3. All elements of an M2M networking platform should be able to be easily integrated into high-level, customized IoT customizations, to aid (rather than hinder) the optimal administration, maintenance, and management of the network as required by the particular vertical market it serves. At the highest central administrative layer, smart grid IoT solutions will share very little in common with intelligent traffic systems, while solar farm solutions will be distinct from both. IoT networking platforms must not intrude on the work of building the final solution envisaged by the customer, but should assist in achieving that goal in every possible way.

4. An IoT networking platform must make the connectivity layer as transparent as possible, effectively turning the intermediate portion of the network between the edge and the core into a black box, with which system integrators and application engineers never need concern themselves.

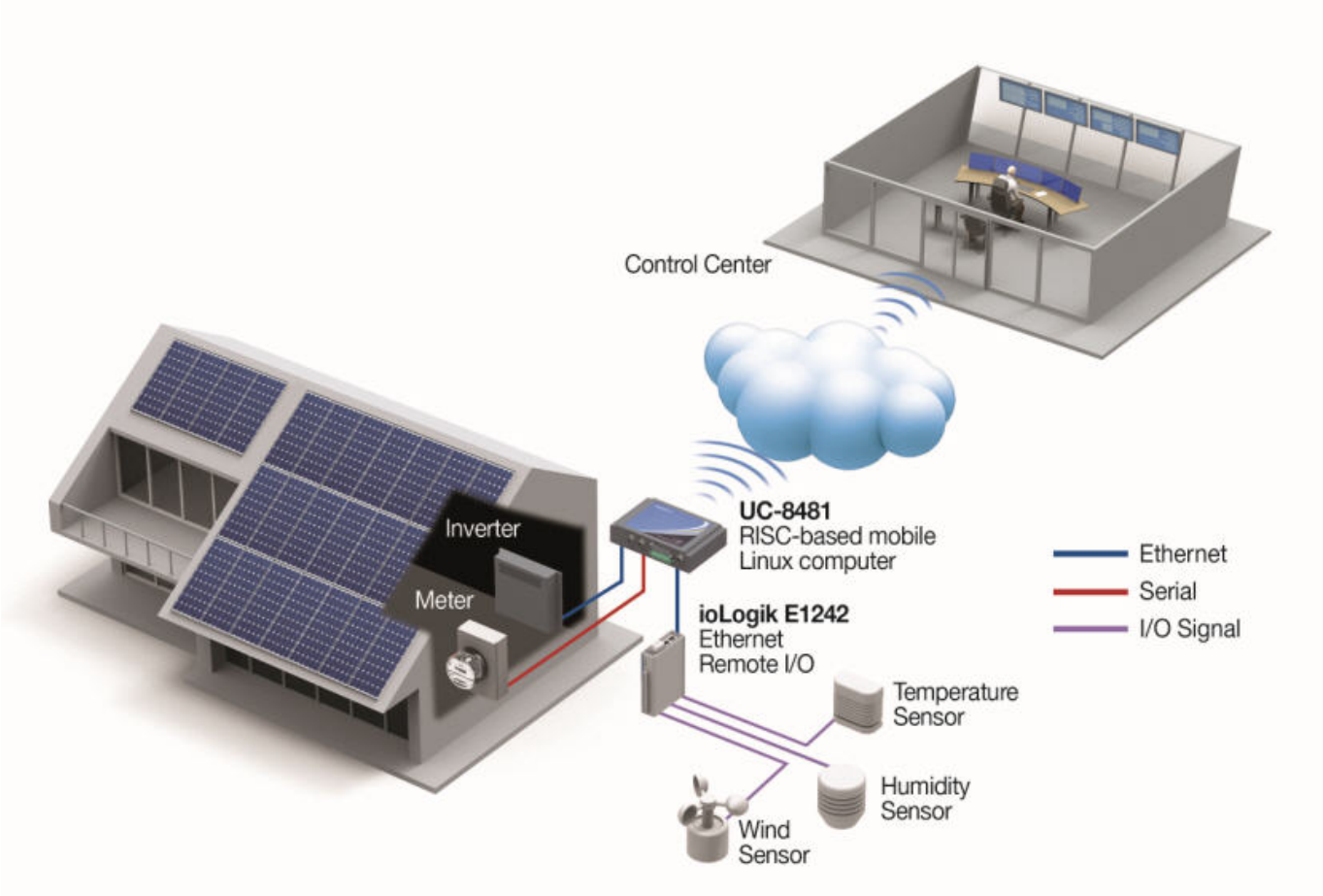
5. Communications between the edge and the core must reliably process all data, regardless of the health of the network as it is accumulated. This means asynchronous, encrypted transmissions between the edge and core, with strong failsafes to guarantee the physical integrity of the data.

As these considerations reflect, any computing platform (or other networking

device) intended for use in IoT deployments should be engineered from a system-wide perspective, where each device is viewed as part of a mutually-supportive interlocking whole, rather than as a single, one-off network tool. A recent solar power deployment for the smart grid is useful as an example, here. Consider a solar station where an embedded computer is used as a gateway and station controller for a residential solar generator system. In this system, the inverter, meter, and remote I/O system are all connected to an embedded RISC platform that manages local automation and data logging, while maintaining remote wireless communications with the central control station. This residential system must therefore maintain network communications over serial, Ethernet, and fieldbus interfaces, as well as either 802.11 or perhaps cellular wireless.

By way of Example…

As IA engineers know all too well, when forced to work at the lowest programming layer the configuration of remote I/O gateways (for automated alarms and event-triggers) is a laborious task. Similarly, setup of fieldbus devices (using, say, Modbus) or wireless addressing and failovers can be equally toilsome. This is where pushing automated features out to the very edge of the network is most useful, and can deliver the most cost-effective solution. Engineers want all of these basic tasks reduced to simplified steps that may be bunched together or automatically initiated using a single, common process. In this way, masses of edge devices may be efficiently and rapidly configured with a dramatically reduced need for user input.



Moxa software is an excellent example of how platforms with integrated, automated setup and administration utilities put the work of configuration and maintenance well into the background of system design and deployment. MXconfig is a mass configuration tool that speeds up the configuration of 100 switches by a factor of ten. MXview uses SNMP to automatically discover, query, and configure edge devices (including the setup of OPC 2.0 tags), and then automatically assembles the results into a visualization of the wired ring. At the same time, Synmap, the virtualized process monitoring and control interface, also utilizes SNMP to serve as a universal control protocol that may be used to script any device that supports it, without any need for further compiling or low-level adjustments. Smart Recovery is a fully automated, BIOS-level system re-write utility that allows administrators to both trigger remote rewrites of the entire software system, or to configure remote devices for fully automated recoveries at either scheduled times or critical events. Because Smart Recovery operates at the BIOS level, it is even capable of restoring a system that has become so corrupt it can no longer boot up. Moxa's DA-Center automates the setup and administration of databases, easing the conversion and display of field data and simplifying connectivity setup with edge I/O, while Active OPC Server delivers asynchronous, event-driven push communications — from the edge to the core — for remote I/O devices, while enabling DHCP addressing on remote ioPAC units. For cellular wireless connections, OnCell Central Manager and Guaranlink ensure that cellular nodes with hidden addresses may be set up for direct communications with the core, while also providing network association fallbacks in the case of station failures.

When combined with modular computing platforms like Moxa's DA series of rackmount servers, or the UC series of universal computers for embedded and edge solutions, these software enhancements allow for secure, reliable, highly automated deployments of massively distributed networks like those now envisaged by smart grid solutions providers, traffic systems engineers, or residential solar power providers.

These are early, strong steps towards creating a virtualized connectivity layer specifically engineered for industrial cloud solutions. The automation involved vastly simplifies the deployment, setup, and management of industrial networking and edge devices, while consolidating and simplifying their management at the central core. By calling upon tailored software solutions carefully integrated with key hardware optimizations in networking, I/O, and embedded computing platforms, industrial cloud engineers will be able set aside the work of connectivity integration and low-level coding to concentrate on the work of developing the most effective system for their needs.

Source URL (retrieved on *03/07/2015 - 12:30am*):
http://www.foodmanufacturing.com/articles/2014/05/building-industrial-cloud