Embedded Controller and Firmware for LXI: Why reinvent the wheel?

By Stefan Kopp
Software Development
LXinstruments

In one form or another, embedded controllers are found in every LXI instrument. Often seen as a mere necessity (not a valuable feature), the development effort going into the controller and its firmware is frequently underestimated. This article takes a brief look at the typical instrument firmware structure and then discusses an alternative to building your own embedded platform from scratch: reusing existing (off-the-shelf) firmware building blocks.

LXI: zooming into focus for smaller players

Since its inception in 2005, the LXI standard has seen overwhelming success in the market. The standard was initially driven by major players who had already been offering LAN-based instruments for some time. Such companies typically design their own platform solutions for instrument connectivity and reuse them for various product lines.

Nowadays, with LXI becoming ubiquitous, more and more smaller, specialty manufacturers desire to jump on the bandwagon. After all, the power of Ethernet is attractive to everyone, and embedded controllers with Ethernet are available at low cost. Why not do it right in the first place, and design the instrument according to the LXI standard?

Embedded platform and firmware: often unloved, often underestimated

While tailoring and tuning a general-purpose embedded platform for use in an LXI instrument is doable, it is a considerable investment, nonetheless. Often the protocols required (such as mDNS or RPC for VXI-11) are already supported – still, firmware is required to orchestrate these elements in the right way to result in an instrument which offers a pleasant user experience and conforms to the LXI standard.

Time to market is a key concern for all manufacturers, including specialty suppliers. How do you avoid the embedded controller part becoming a headache and delaying your product introduction? The answer can be quite simple and you might have heard it before from similar development situations: reuse existing building blocks and focus your R&D team on the measurement part itself.

Getting more specific: firmware structure

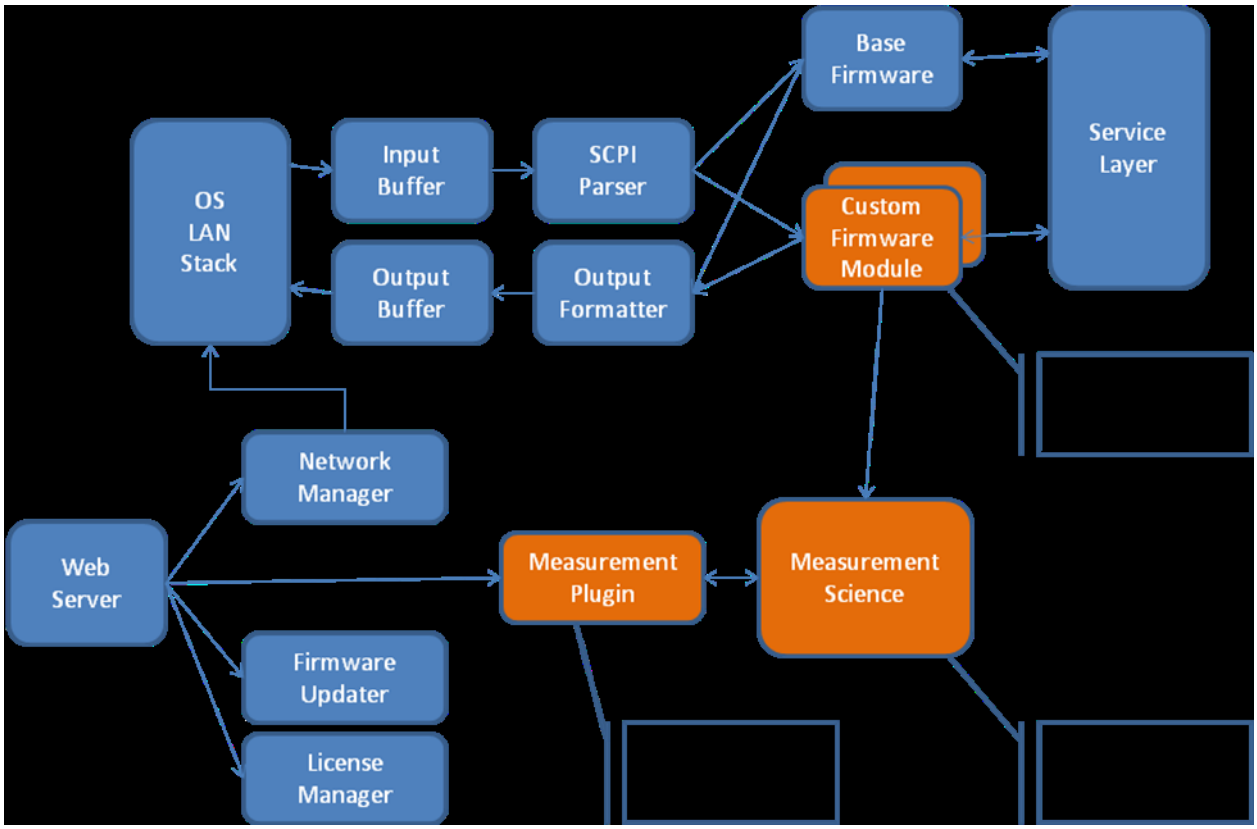

Fig. 1: Simplified Block Diagram of Typical Instrument Firmware

The above block diagram depicts the typical firmware elements found in modern instruments. Let's quickly review the major elements:

- The *SCPI Parser* processes incoming commands and manages the associated command handlers. For example, commands which deal with network management, instrument status etc are handled by the base firmware, whereas commands which exercise the individual instrument functionality are handled by one or more custom firmware module.

- For synchronization and access to common data, the firmware elements access a *service layer* via an application programming interface (API). For example, all modules may report errors into the instrument error queue, which, in turn, will automatically update the status system (condition registers, status byte).

- The *custom part* of the firmware will typically control the measurement hardware of the instrument using an interface such as digital I/O (GPIO or parallel bus), I2C, dual-port RAM etc (depending on the bandwidth requirements).

- The *web server* provides interactive access to instrument status information, network configuration and, optionally, to some or all of the functions supported by the various firmware modules.

While the above described functions would be sufficient for LXI certification, modern instruments frequently offer useful additions such as *firmware update* and *license*

management (for option activation) via the built-in web server. Also, many instruments offer USB connectivity in addition to LAN.

What is quite obvious from the diagram is this (note the colors chosen): while the firmware is quite complex, most of the elements are standard (shown in blue). With a clean, well-designed structure, the customization work is limited to the custom firmware modules where the individual measurement science is controlled (see orange elements). There is no need to modify the common firmware infrastructure itself.

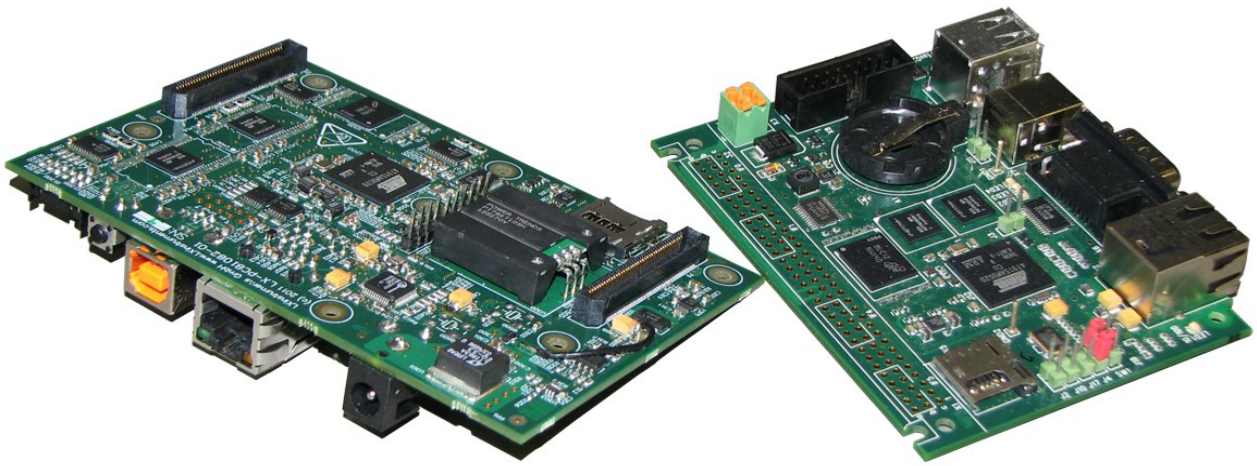

Fig. 2: Controllers
2 generations of the LXinstruments LXI Controller. The new version 2 of the controller (left) offers connectivity for 2 specialised function PCB's for almost any type of product.

Using off-the-shelf components

Today's world of embedded computers offers professional development tools, many of which are open-source (which, among other things, translates into long-term availability, an important aspect, given the long product life cycle we often see in the test market). For example, tools widely used for embedded development include the Eclipse development environment and the GNU compiler tools.

Combining these tools with prebuilt firmware components helps shorten the firmware development time and allows you to focus on your core area of expertise: the measurement science. LXinstruments, a long-standing member of the LXI Consortium, offers embedded solutions, very similar in design to what is shown above. These solutions are found inside of several certified instruments on the market today and are also used by a number of manufacturers who decided not to go through full certification (but keep that option for later).

One such company is BitifEye, a German company who has been offering test software, test accessories and test services to international customers for many years and is just now introducing a new solution for automated HDMI cable testing. The solution comprises an NWA with opt. TDR and ECal module, an optional switch and the BitifEye ValiFrame Test Automation Software. The HDMI cable under test is connected to the switch by two access adaptor test fixtures. According to Alexander Schmitt, founder and director of BitifEye, the use of the LXinstruments solution framework and SDK was instrumental in

getting their firmware market-ready in time while offering advanced features such as web-based firmware updates and licensing.

Summary

Even though LXI is all based on industry standards, firmware design is a big task. There is room for individualism (the mandate of the LXI standard leaves flexibility in many areas), but there is also the risk of unforeseen issues and unexpected design cycles. Using off-the-shelf components allows you to take the fast lane to firmware development, without sacrificing functionality and proper design. This increases the attractiveness of LXI especially for smaller players who find it difficult to justify the development of advanced firmware features from scratch.

About the author: Stefan Kopp is part of the test&measurement world for more than 20 years. He accompanied the birth of the LXI standard very closely from a dedicated position of a founder company. Since 2010 he is in charge of the Software Development department at LXinstruments. info@lxinstruments.com