

Field Report: Industrial Control Remote Lab

Two engineers provide an objective evaluation of the Remote Lab for an Intel® Atom™ processor-based industrial reference platform

Review by Joachim Meyer and Michael Rückauer
Karlsruher Institut für Technologie (KIT)

The Remote Lab at Intel allows engineers from around the world to evaluate Intel® Embedded Platforms via the Internet. Two engineers from the Karlsruher Institut für Technologie (KIT) review the Remote Lab and accompanying tutorials for an industrial reference platform based on the Intel® Atom™ processor.

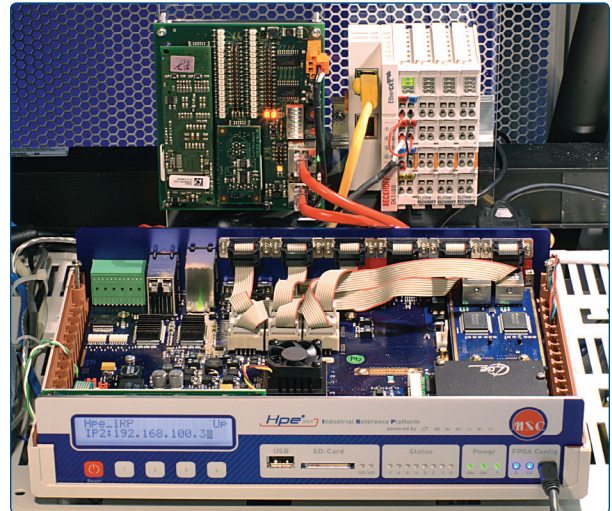
First Impression

We started by logging into the Remote Lab with a preliminary version of the “Remote Lab Access Guide” (RLAG). We had no problem following the guide to build the VPN tunnel and to create all the necessary connections. Unfortunately, the webpage, begw1.mcplab.net, was not signed properly, which resulted in security warnings. Aside from that issue, all the other parts in the RLAG worked very well, especially the webcam feature with its pre-stored locations, which was interesting.

Tutorial 1 – Introduction to the Industrial Reference Platform

After getting familiar with the capabilities of Remote Lab (e.g., ssh login, KVM, CC, etc.), we started with the first tutorial. The goals of this tutorial were to introduce the Hpe® Industrial Reference Platform (IRP), present its Linux® environment and demonstrate how to create and use an FPGA design with the help of the HPE Desk software. After describing the IRP, the tutorial covered the use of basic Linux functions, such as loading and unloading kernel drivers, as well as some custom features, like examining the FPGA design using “lsfpga.” The tutorial closed by describing how to use the Hpe desk software to recreate the FPGA design. We found the usability of the `oss_rom` in combination with the “lsfpga” command particularly interesting.

From our point of view, all of the goals were met. This tutorial was useful for imparting the basic knowledge needed to work with the IRP and move onto the more complex tutorials.



Tutorial 2 – Hardware-based PCI Express Performance Measurement

The main goals of Tutorial 2 are to analyze interrupt latency and use the results to minimize it. Additionally, it covers user mode driver development for accessing custom hardware modules on the FPGA from the Linux system. This is done in a similar way to Tutorial 2. First, an FPGA design, including all necessary hardware modules, is built. Next, the user is guided through a step-by-step measurement sequence with different options and environment settings. By analyzing the plotted results, it's possible to see how different configuration options impact interrupt latency.

Low interrupt latency is crucial when responding to critical events on time. In our eyes, it is valuable that beyond presenting results, the tutorial includes the corresponding software code to implement the functionality in custom drivers. Thus, a good compromise in explaining details and providing references was found. Altogether, this tutorial provides a basic insight into driver development and interrupt-handling for the IRP system.

Tutorial 3 – Hardware-based Interrupt Latency Measurement

In Tutorial 3, users measure the maximum data transfer rates over the PCI Express* link between the Intel Atom processor and the FPGA. First, a suitable FPGA design is set up containing the FPGA core, memory systems, DMA controller and a trace module that logs accesses to the AHB bus. Subsequently, various read/write access tests are performed from a predefined benchmark application. The objective of this tutorial is to show the relation between the theoretical and practical performance of the PCI Express link. In addition, it explains the observed performance gap and gives hints about how to improve the actual performance, which benefits applications with heavy communication traffic.

The tutorial provides basic analysis techniques that can be employed with any design. In particular, it demonstrates how to use the hardware trace module to generate statistics about PCI Express performance without influencing the actual application. A nice feature is the ability to create well-arranged plots from the raw trace data. We felt the tutorial provided a good starting point for analyzing a custom design.

Conclusion

The tutorials are relatively easy to complete because the instructions are clear and supported by screenshots. Tutorial 1 provides basic knowledge required by the other tutorials, so its technical level is rather low, presuming users have some knowledge about Linux and FPGAs. Important nonetheless, the tutorial provides an introduction to the IRP system and the associated software. The other tutorials cover more advanced topics, which tackle essential questions related to system design using the IRP.

In our opinion, the Remote Lab, together with the provided tutorials, provides a good opportunity to evaluate the features and the performance of real hardware before actually selecting the system. In addition, the techniques presented in the tutorials establish a good starting point to analyze and optimize your own custom designs. The Remote Lab gives the user full control over the IRP system, like having it in your own lab. Therefore, beyond the tutorials, the Remote Lab presents many more possibilities to evaluate the IRP system and determine whether it fulfills the developer's requirements.

For more information on the Remote Lab, visit <http://edc.intel.com/Platforms/Remote-Labs>.

