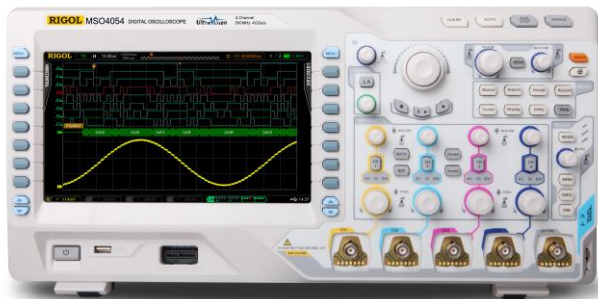# Debug and Analysis Considerations for Optimizing Signal Integrity in your Internet of Things Design

MSO4054 Mixed Signal Oscilloscope

Introduction

As the Internet of Things continues to expand to include applications including home automation, fitness, video, and tracking as well as traditional embedded electronics use cases the needs for testing and optimizing these designs become clearer. Even as IoT expands, the test requirements have begun to crystallize into a set of capabilities that can assist a design team in achieving their goals. This is especially important given the smaller, more versatile teams and companies charting a path in IoT. As companies and engineers often new to large scale design work attempt to make new types of successful consumer products, IoT projects will continue to progress in fits and starts until designers find the optimal mix of form and function for each target application. Too much focus on form inherently creates reliability and annoyances like additional charging time. By first considering test requirements as a method for evaluating these concerns, a design team can speed time to market and ultimately reduce the iterations required to create a successful product platform. Selecting the best test equipment for the task at hand also enables engineers to stay within their startup budget requirements. Signal integrity is a key design aspects whose importance is magnified by these inherent tradeoffs in the IoT landscape. Focusing on signal integrity as a key indicator of the optimal approach for an IoT application from debug through final design enables a small IoT design team to leverage their capabilities and ultimately reward their customers with performance and ease. Here we will be discussing important examples of signal integrity test methodologies for IoT devices and how they impact the user experience from form to function.

Figure 1: Basic SPI analysis with analog oscilloscope channels

Form and Reliability

It is crucial to consider how dramatically signal integrity and reliability are affected by mechanical implementation. The trend for IoT products, especially wearables, continues to be sleeker, lighter, and smaller. Optimizing the form of the product for users is therefore often in conflict with reliability, ruggedness, and signal integrity requirements. Designs that strike a balance too close to the aesthetic ideal often fail to succeed for these core functional reasons. With more demand for waterproof and rugged designs that also integrate accelerometers and other sensors we create an environment on our device where signal integrity can often be compromised. This can lead to shorter battery life, user feedback delay, and even critical data or system faults.

Let us test a few examples looking for issues in communication on our SPI peripheral bus. The SPI bus is a common communication protocol for accelerometers, GPS chips, and many other sensor and actuators. Continuously monitoring a design to make certain reliable communication will be maintained to quality standards is an important test technique for this type of product development. In these tests, we will provide several methods for verifying and comparing SPI communication that can be used iteratively as your design approaches completion.

First, there are several aspects of serial communication that can are commonly affected by connections, layout, and stress or other aspects of long term use. These include bandwidth, noise, and impedance. Again, our first goal is to establish both a goal and a baseline for these parameters. The first step is to monitor and verify these signals with the analog channels of an oscilloscope as shown in Figure 1.

Figure 2: Pass / Fail Analysis to highlight crosstalk on a communication bus

Figure 3: Comparison with mixed signal channels for data interpretation

In this test, we can see that our connections appear to have more than enough bandwidth but we can also see some crosstalk noise especially on the data and chip select lines. The green bus values are an internal decode of that signal by the scope. As long as these are stable for each transmission then the noise is likely acceptable. Since crosstalk is a common issue here it can be important to test the bus while other peripherals are being serviced and activated asynchronously. One methodology is to utilize a pass/fail mask, which we have implemented in Figure 2, looking for unacceptable noise levels in a large sequence of traces.

One advantage to this approach is that masks can easily be saved and reused on different board revisions or code revisions within a project. Simple tests can therefore help to quickly identify the version or build that first showed an increase in noise or change in performance. It is important to internalize that the opposite can be true as well. Device performance may be so reliable that different layouts or smaller wires for connections may be possible creating more freedom for the overall mechanical design. When significant data errors or bugs do come into play another verification method can be used. To get a clearer image of how our IoT platform is interpreting this data we can use our mixed signal channels for comparison.

The digital channels on the oscilloscope (shown on the top of Figure 3) can be set with threshold values that mimic the bus controller on the IoT platform. The device does not see the analog signals, but interprets their digital equivalent. Therefore, when having data issues, it is important to look at both the digital and analog representation. This enables engineers to quickly discover where data failures are occurring as well as the analog root cause or noise source that may be contributing.
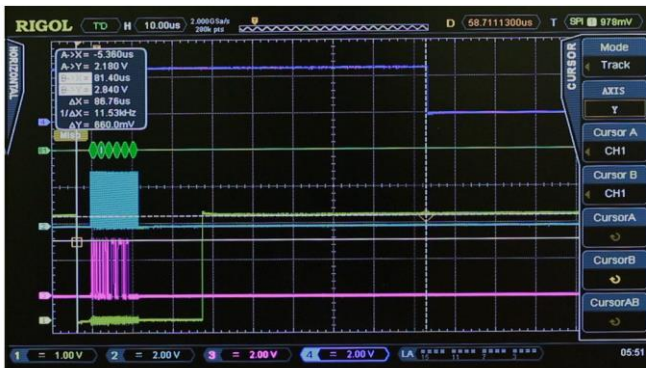
Figure 4: Latency measurement from SPI bus to LED on our IoT development board

Lastly, one important consideration when implementing sensors in an IoT device is the latency between bus communication and action. Of course, this is highly dependent on both the platform and the code methodology and can change dramatically. Latency is the time it takes for the system or platform to interpret the peripheral data and take an action. As a latency baseline, we ran a simple test that toggles an LED when a SPI read is completed. The cursor measurements on Figure 4 highlight this.

From this baseline on your platform you can model how code changes impacts this latency and then you can base decisions about servicing these sensors based on the balance of use case and reliability requirements with battery life and implementation.

Latency, noise, bandwidth, and impedance all effect signal integrity and reliability. These key measurement techniques can be used throughout development, use case, and reliability testing to optimize the overall design and anticipate failure modes. As part of an overall IoT design strategy, continuous evaluation of signal integrity established baselines and limits helps speed time to market and customer acceptance with small impact to equipment budgets.

Conclusions and key learnings

When working in the fast-changing atmosphere of IoT design and development reliable test methodologies become increasingly important. As engineers integrate the newest sensors and platforms on the fly to reach highly competitive markets as fast as possible, understanding core customer requirements and tradeoffs and how those can be evaluated and compared throughout the development is an important step toward improving the strategic design process.

Whether the challenges of an application are more form or function, issues related to signal integrity and reliability are fundamental elements of design in the IoT ecosystem that play a significant role in market success. Establishing these principles early in the process and testing them iteratively is one of the best ways to limit budget and schedule risks in the latter stages of the design. Modern, easy to use test equipment that is more affordable than ever can be utilized to develop the limits and baselines that will guide an engineering team through a successful product development.

For more information on our instruments and solutions, please go to rigolna.com or contact us directly at help@rigol.com or call us toll free at 877-4-RIGOL-1.

RIGOL Technologies USA
10200 SW Allen Blvd, Suite C
Beaverton, OR 97005
877.474.4651