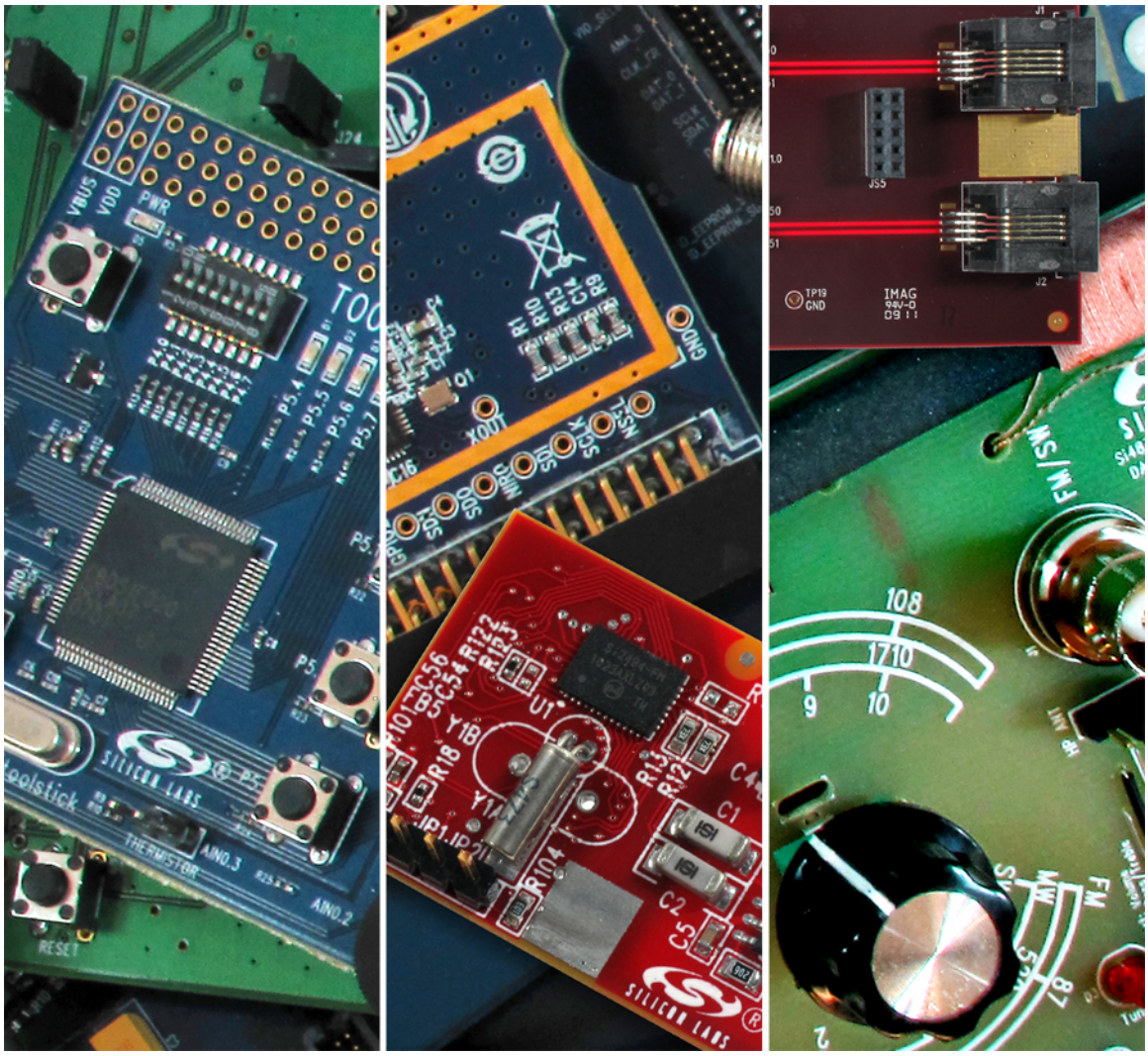

How to Overcome Technical Hurdles

During MCU Development

SILICON LABS

Engineering for a Mixed-Signal World.

Introduction

An embedded systems developer starts a project with a blank computer screen and emerges from his or her office a week later with a product that's perfectly functional and ready for prototyping. The developer's code compiles without even a warning, and their schematics produce not a single design rule violation.

One of the most useful skills an engineer can acquire is to develop an intuition for where the root cause of a problem might be located.

Then the embedded developer wakes from his or her blissful dream, goes to the office and faces reality: the IDE stubbornly repeats the same cryptic compiler errors no matter what code changes the developer makes. The developer's board designs, already long overdue, still have a discouraging "work in progress" look, with disconnected power nets and routing that would confuse Rube Goldberg.

The axiom "nobody knows everything" has never been truer than it is in embedded systems development.

Developers scramble to gain expertise with a collection of new-to-them integrated circuits (ICs) and conflicting use cases as product deadlines whiz by and marketing-driven feature creep piles on more and more features to a project's already ambitious specifications.

Fortunately, help is on the way. IC vendors – in particular, microcontroller (MCU) vendors – provide a wealth of online self-help resources geared to embedded developers. Additionally, these vendors staff teams of capable engineers that can answer your questions and help you overcome the challenges you face during the system design process. You just have to know where to look for help and how to ask for help effectively.

The First Step Is Admitting You Have a Problem

The most frustrating part of problem solving can be developing an understanding of what you do not know. One of the most useful skills an engineer can acquire is to develop an intuition for where the root cause of a problem might be located. Even if the developer doesn't know the solution, he or she must be able to "draw a box" around the problem to some extent. Simply saying "it doesn't work" and throwing up one's hands will only lead to more stress. We all benefit from the fact that we operate in a deterministic world. The problem being observed is caused by one or more factors in the system being developed. Take a deep breath, procrastinate on the Internet for a bit and then dig into the solution.

Self-Help Resources for MCU Developers

In addition to the requisite product datasheets, IC vendors provide resources such as user forums, knowledge bases, application notes and reference designs. They create these resources out of a primal survival instinct. Every customer that finds an answer autonomously is a customer for whom staffed engineers will not have to expend any energy and time helping directly. This frees up company resources to generate more self-help resources as well as enabling staff engineers to make

progress on all their other responsibilities, such as validating new designs and generating collateral for soon-to-be released products.

Developers might be surprised at the breadth and depth of the resources available to them, as well as the frequency with which their exact issue has been explored. The accuracy of these resources is no accident: most of these articles and documents get written after a developer like you contacts technical support. Engineers on staff monitor incoming questions and convert helpful answers to documents and knowledge base articles so that the next developer facing the same issue will have easy access to that information. This positive feedback loop ensures a constantly improving and refining digital curation of answers just waiting to save the day.

Developers might be surprised at the breadth and depth of the resources available to them.

Harness the Power of the Hive Mind

Most vendors maintain user forums for their customers, and the best user forums in the semiconductor industry provide insights into vendor products that rival or outmatch the company's own internally generated resources. Engineers are problem solvers and tinkerers by nature, and this nature is never more evident than when they're given the chance to communicate with each other en masse. Vendor-internal engineers also tend to moderate forums themselves, and they will step in with helpful posts when possible.

Visiting a user forum can be an experience that is as humbling as it is encouraging: there are so many brilliant people working in the semiconductor industry. In many cases, you might find the answer to your question after a quick search of the forum. Otherwise, you should consider posting your question to these amorphous blobs of technical know-how.

A vibrant user forum's collective level of technical expertise asymptotically approaches infinity. That said, be prepared to field a few off-the-wall questions in response, and take it as a given that your fellow forum- visitors are all well-intentioned and as thoughtful as you are. Also keep in mind that the shaping of your question is crucial to getting helpful answers. This brings us to…

How to Ask a Question

The importance of "drawing a box" around a problem was previously mentioned. Another way of describing this part of the process is to try to discern the components of your project that are relevant to the problem's cause. Look at it as a signal-to-noise ratio where the more information you include in the problem statement adds a steadily higher noise component, making it increasingly more difficult for others to find the signal (the problem's cause).

Try starting with the complete universe of causes, meaning the system as a whole. Write down a list, if it helps. Then go through that list as systematically as you can and strike items that seem irrelevant after careful examination. However, just remember to strike items "in pencil," allowing you to change your mind. You don't know the problem's cause, after all. Be prepared to provide more information about

those components you initially dismissed as unrelated as dialog with others progresses.

Be as forthcoming with information as you can (and as any NDAs or other restrictions allow). If the problem has a substantial firmware component, consider including code snippets in your problem statement. If the problem has the look and feel of a hardware-related issue, try to find a way to include a portion of your schematic. It helps to understand that asking a question without providing relevant information is a little like going to the doctor's office with an ailment and staunchly refusing to allow the doctor to examine you. People can only help if you ask for it, and people can only be substantively helpful if you ask your question with an adequate degree of specificity and clarity.

Understand that applications engineers and sales representatives are interested in answering your question.

Support Staff Engineers Get Paid to Help You

Whether you end up submitting your question to a field applications engineer, an applications engineer or even a sales representative, understand that all of these people are desperately interested in answering your question. Their primary motivation is obvious: they are engineers just like you, and therefore they feel a deep satisfaction in making something work that was once broken or making something work more effectively.

The secondary motivation might be nearly as obvious: it is their job. Just as you are evaluated based on your project's relative success, these engineers are evaluated based on their facility at providing the help you need. Take advantage of what they have to offer!

Conversely, it helps to appreciate that as all-consuming and soul-crushing as your problem might feel, your question is most likely only one of many questions that vendor-staffed technical support engineers are trying to answer every day. Be sure to communicate any urgency driving your question, but also understand that it might not be practical to get a response from support immediately after that question is submitted.

If you haven't heard back from a vendor's support staff in a reasonable amount of time (whatever you deem that amount of time to be), feel free to "ping" tech support and ask for a status update. If the answers provided by support don't seem to address your problem, it will behoove you to exercise a bit more patience. Take a step back from the conversation and examine all the information that has passed between you and the support engineers. If you suspect that the support engineer has missed a crucial bit of information that you provided, reiterate that information. If the support engineer has provided an answer that seems unhelpful or irrelevant, ask the engineer to elaborate on their response. It is a certainty that the support engineers are going through the exact same process on their end by reviewing the complete history of correspondence, re-examining all the information provided in both directions and always trying to dig a level deeper to find the problem's root cause.

We’re All in This Together

The technical relationship between developers and vendor resources depends nearly exclusively on effective communication. Search for answers, shape questions with clarity and purpose and demand the attention that your project deserves. When your development project is at its lowest point, always remember that the problems you’re facing most likely have solutions. The more adept you become at finding those solutions using the resources provided by your vendors, the more effective and efficient you’ll become at delivering great designs on schedule. Learn about Silicon Labs’ MCU solutions at www.silabs.com/mcu.

# # #

Silicon Labs invests in research and development to help our customers differentiate in the market with innovative low-power, small size, analog intensive, mixed-signal solutions. Silicon Labs' extensive patent portfolio is a testament to our unique approach and world-class engineering team. Patent: www.silabs.com/patent-notice