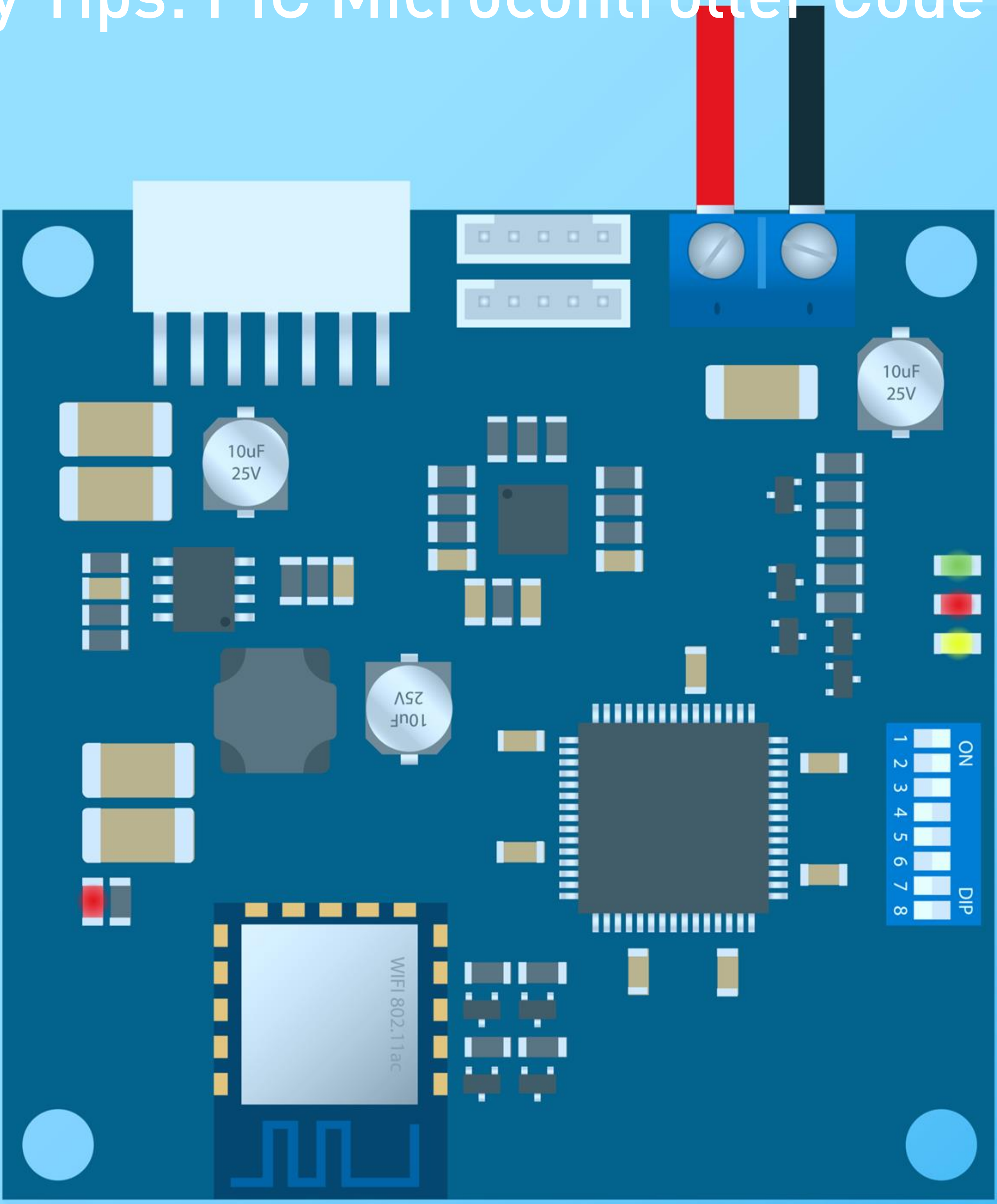
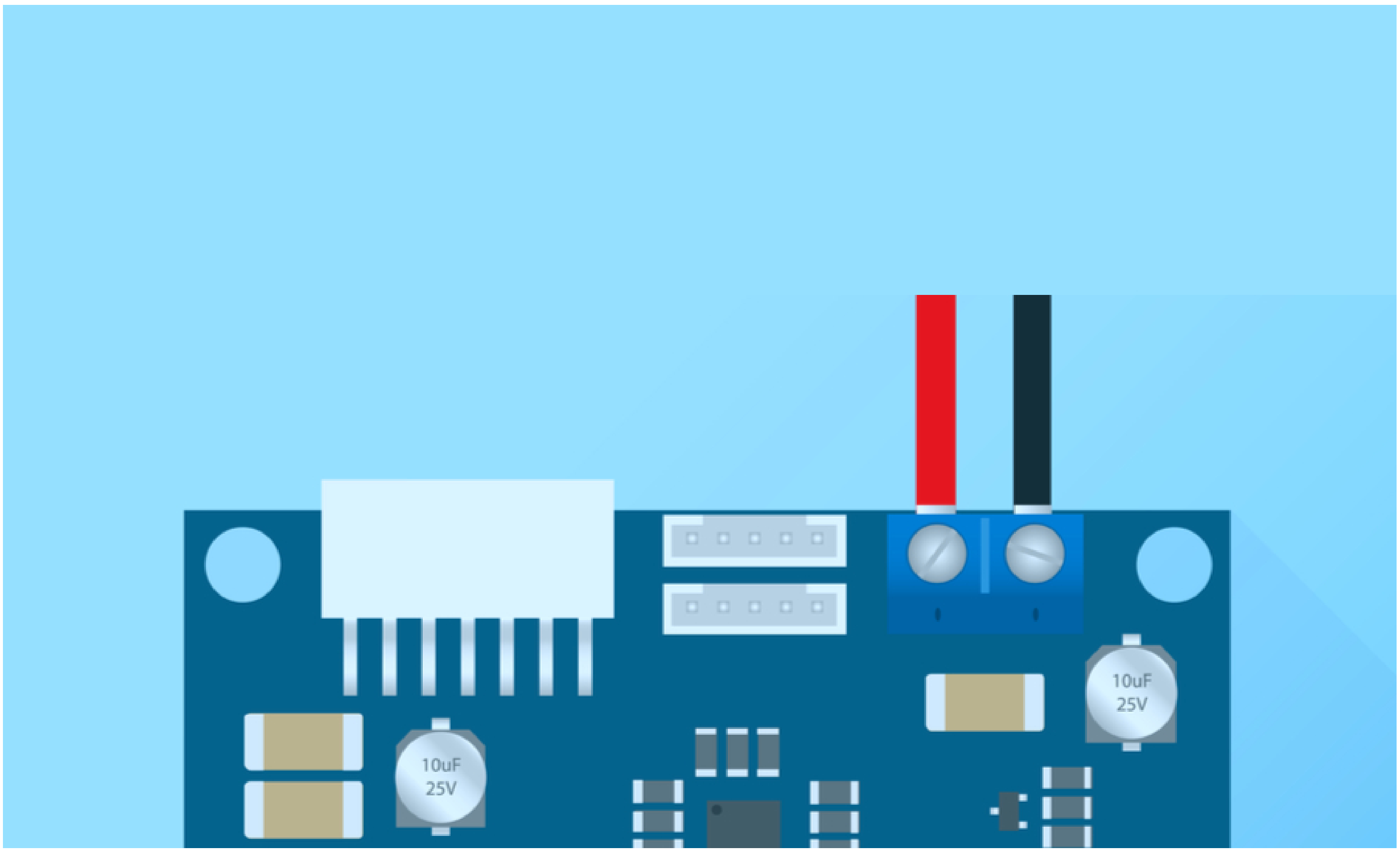
Altium®

Key Tips: PIC Microcontroller Code





KEY TIPS: PIC MICROCONTROLLER CODE

In the past, many PCB designers would shy away from using autorouters in their designs due to their penchant for difficult setups and questionable results. This led to many of us opting for manual hand routing to avoid the trouble and cleanup we'd come to know. However, recent developments in auto-routing technology have flipped the script, allowing for a lot of saved time and effort in your design process. While some of the more critical aspects of your design will still require manual routing, the ability to set the design rules the autorouter will use for your traces will save you the headache.

Not 100% convinced? Join us as we discuss a variety of topics to help you decide if Key Tips: PIC Microcontroller Code is for you, including:

- Give Them No Quarter: Preventing PIC Microcontroller Code from Being Duplicated
- Hide and Seek: Key Serial-Number Chips for PIC Microcontroller Code Protection

GIVE THEM NO QUARTER: PREVENTING PIC MICROCONTROLLER CODE FROM BEING DUPLICATED

Copying, from all understandings, seems like a disingenuous and sub-par strategy for any solution you need; from taking a test in school, to catching up with your friends, it is rare to find a moment where you would be rewarded for simply copying a message or answer. Imitating, on the other hand, is an encouraged practice in most professional industries - follow the guidelines and roadwork laid out before you so as to avoid the tedious work of reinventing the wheel for any given task. If copying is punished, and imitating is rewarded, then what's the difference between the two and how do you protect yourself from incurring future punishments?

From my understanding, imitating is a practice which is there to help you get to a point where you can develop your own strategies, habits, and ideas - not to keep you within certain parameters forever, but to give you an easier understanding of the professional world that you are working in.

But while you are concerned over the debate between copying and imitating and the ethics of each, there are others who may be less concerned about this problem. Take pirates for example. A pirate's life may not be the life for you, but that won't stop one from stealing treasured code from your electronics. If you have an electronics product that is strong or selling well, code copying is a hard reality that you need to address with your product. Especially within electronics industries, code piracy from an unscrupulous competitor will hurt your business. Give no quarter to your competitors, and learn how to protect your code.

DON'T IGNORE THE PARROT: USE PIC MICROCONTROLLER CODE PROTECTION

A great part of my engineering career involved designing with PIC microcontrollers. Protecting my code from being copied had not been a priority until I started a design business; there's simply too much that goes into the process of PCB design and manufacturing for me to have been concerned over one part of this process being compromised. But, after I listened to horror stories of how some electronics businesses suffered huge losses due to piracy issues, I said scupper that and began taking preventative measures in protecting my products.

Reading an unprotected firmware from a PIC microcontroller is relatively simple. You just need to ensure the microcontroller is connected to the programming header and the raw code can be extracted to a computer via the programming adapter. The resulting file is a hex file that can be easily injected into duplicated hardware. But there are many ways to which you can add protection to this code which will heave any would-be pirate.

No reason for not turning on the code protection bit

WANT TO PROTECT YOUR CODE? INSTALL MORE CANNONS!

Code protection should be one of your design priorities from the start. There are various ways to protect your code in your final products and varying accompanying degrees of difficulty to break. Here are a few:

Enable Code Protection Bit: In the PIC microcontroller, there is a code protection bit that will prevent read operation on the program memory when it is asserted. Turning on this code protection feature is as simple as including a directive in the program before it is being compiled and downloaded to the microcontroller. This is the easiest way of code protection but there are suggestions that there are illegitimate methods that may bypass the security features.

Unique ID Authentication: Instead of preventing the code from being read, this method requires the application to verify itself against a unique ID before launching the main program. It usually requires manual insertion of a secret identification number during the first startup before storing the unique ID from an external device to a secured storage. In the case of the code being copied to another hardware, the firmware will not be able to function without the stored unique ID and the secret identification number.

Epoxy Coated: In products where there is no need to update the firmware of the microcontroller, the epoxy coating can be used to physically prevent access to the microcontroller. This is an effective way to not only prevent code piracy but also any attempt to copy and re-engineer the hardware itself. The downside is that the product has limited serviceability if it experienced failure after a period of time.

Self-Destructing Circuit: Just like in certain movies where the characters turn kamikaze mode to protect secrets being tortured out of them, self-destructing circuitry can be used in sensitive applications. I've worked in a highly sensitive industry changing design where the client insists on a self-destructing circuit to prevent any attempt of piracy at all. It uses sensors to detect any illegal attempt to access the hardware and connect the microcontroller to a high voltage source. This is a drastic measure to take and the system may accidentally activate the self-destruct mode if the sensor malfunctions.

Last resort for highly sensitive applications

Take the protection of your code seriously, and you'll be sure to keep your treasure from coast to coast. If any of these methods feel inadequate to you, you can also try the tried-and-true method of keying serial-number chips directly, too. Don't try to pilot your ship across rocky waters, and worry about pirates—make your protection guaranteed. Use great PCB design software that lets you make the most of your designs and guarantee their safety. CircuitStudio comes with a great range of tools and add-ons that can help you work as securely as possible.

Need more help in preventing your microcontroller code from being copied? Talk to the experts at Altium now.

DON'T SETTLE FOR BETTER LATE THAN NEVER: PACKING AND SHIPPING YOUR PCBS

I mailed my last Christmas present yesterday, a few weeks after the holidays are over. I'm only a little embarrassed; I like to do a very precise job with wrapping even if I'm not terribly concerned with punctuality. The package was for a friend overseas who had a craving for her favorite American candies. Let me tell you: a beautiful wrapping job on a bag of fun-size Butterfingers is no mean feat.

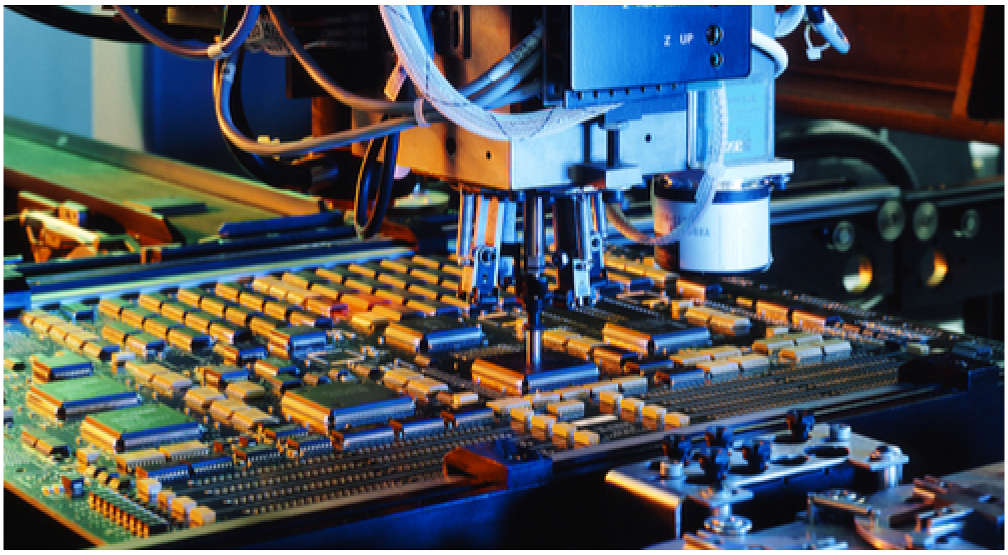
Wrapping and packaging are more than ornamental when you're packing up a PCB. Before you get anything produced by a new manufacturer, you should ask about packaging. You want to be sure that your board is adequately protected. If a Butterfinger is damaged in shipping, you still get delicious crumbs, but a PCB can't always be salvaged.

WHY DOES THE PACKAGING MATTER?

If you don't have an obsessive need for square corners and symmetric tape, then fussing over your PCB packaging probably seems like a waste of time. As long as the boards don't get crushed in shipping, everything is fine, right? PCB packaging is almost as specific as the PCB function. You designed your PCB to protect it from its expected operating environment and you need to wrap your PCB up to protect it from hazards of shipping and storage.

Shipping hazards range from the obvious dropping and crushing, but other mechanical damage can also occur. In extreme heat or cold, boards can warp, solder can creep and crack. You might have someone handle the packages who, like me, shakes boxes very aggressively to find out what's inside.

Less obvious damage sources are the same things you protect PCBs from: moisture and electrostatic shock. It's possible to get any, all, or none of the types of damage to your boards during shipping. It just depends on how well they're protected and how aggressively they're shaken.


Try not to shake your manufacturing machinery.

WHAT TYPES OF PACKAGING ARE THERE?

Packaging varies based on what you need to protect against. The protection is not always obvious just from looking at the wrapping, so it's helpful to have a little background:

Tissue paper and bubble wrap: Unless you're shipping bare PCBs (which just have a rubber band holding them together), this is probably the most basic packaging you'll see. This is fine for very simple, robust boards, but if you have more sensitive components, you probably want to step it up.

Pink poly: Pink poly is a type of plastic wrap that provides ESD protection. It comes as plastic wrap, bags, and bubble wrap. In addition to the cushion of a pink poly bubble wrap, it protects against shocks that occur when the box full of boards is packed and unpacked. You should be aware that the material is not compatible with certain plastics, especially polycarbonate, so select your materials accordingly.

ESD Bags: Many ESD sensitive components and some larger hardware will get shipped in ESD bags. They are shiny, silver plastic, and often have a Ziploc style seal. You can get these in various sizes, and have your ESD sensitive boards packed into individual bags to keep them safe from static shocks during transit

Moisture barrier: If your PCBs are moisture sensitive (most are, at some level), or will be shipped by ocean freight, or through a humid climate, look at moisture barrier packaging. Your PCBs will be individually sealed, probably with a little packet of desiccant, to keep out moisture in the air, regardless of where the boards travel en route to you. Moisture barrier packaging is usually puncture resistant, too, so it gives a little extra protection to your boards.

Vacuum sealed: For the hardcore moisture sensitive applications, you can have all the air sucked out of the packaging around your PCB. I think it looks like a neater version shrink wrap, and of course, I prefer everything neat and tidy. You can also get vacuum sealed ESD packaging if you need protection on multiple fronts.

Vacuum sealing is used for food, electronics, and anything else that needs to be protected from air and moisture during shipping.

WHAT DO I NEED FOR MY PCB?

Choose a packaging that's going to provide adequate protection for your board. After designing and manufacturing, you don't want the boards to get damaged in transit. If you are receiving multiple boards in a single shipment then have them packaged based on their custom requirements. I wrap a drone very differently than a bag of candy bars, but I want them both to arrive unscathed.

When you're getting PCBs manufactured ask what the standard packaging is. Find out about pricing for other options. If pink poly bubble wrap is enough, and they usually vacuum seal, you might be able to get a discount. If you have concerns, ask to get a sample of the packing material before they start manufacturing anything for you. If you are on-site for an audit, watch their packing methods, and take a good look at the materials they are using.

While not as fun as picking wrapping paper, making careful decisions on packing will keep all your hard work from going to waste right at the end. On the other hand, a good decision to make right at the beginning is which PCB design software to use. With easy and smart manufacturing outputs, intuitive drawing processes, and strong templates to work off of, Altium's CircuitStudio is a great option.

If you're looking to know more about your PCB manufacturing process, and how your design software can make it easier, consider talking to an expert at Altium today.

ADDITIONAL RESOURCES

Thank you for reading our guide on Key Tips: PIC Microcontroller Code. To read more Altium resources, visit the Altium resource center here or join the discussion at the bottom of each original blog post:

- Give Them No Quarter: Preventing PIC Microcontroller Code from Being Duplicated
- Hide and Seek: Key Serial-Number Chips for PIC Microcontroller Code Protection