

Teuthida Technologies Home

All about embedded systems engineering.

Errata errata!

Posted on [January 29, 2012](#)

To paraphrase (poorly): A funny thing happened on the way to the blog! This article was to have been about the interesting (really!) topic of calibrating a touch screen and some of the unusual twists and subtleties involved with that process. Along the way though, something came up: Errata!

In working on my code, it occurred to me to check to see if there were any errata outstanding on the [PIC32MX460F512L](#) processor I was using. I was a little alarmed to find no less than 68 errata items for this chip. I really should not have been. Given the incredible complexity of the part, it should be off no surprise that 68 defects could be found, and most of them were very obscure. One point did catch my eye however. Here it is from the Microchip docs:

62. Module: PORTs

When an I/O pin is set to output a logic high signal, and is then changed to an input using the TRISx registers, the I/O pin should immediately tri-state and let the pin float. Instead, the pin will continue to partially drive a logic high signal out for a period of time.

Work around

The pin should be driven low prior to being tri-stated if it is desirable for the pin to tri-state quickly.

Affected Silicon Revisions

B2	B3	B4	B6				
X	X	X	X				

Not to get too detailed at this point, but my code fell exactly into the above scenario. I constantly switch between 2 pins to output to generate a voltage slope while the other 2 pins are analog inputs to read the voltage. Since the issue and its fix were from the maker of the chip, I felt safe adding it to my code, even though it seemed to be working fine already. As instructed, I added code to set the pins to “0” before setting them as inputs and patted myself on the back and went on to other parts of the code.

Until it came time for testing... Suddenly, my well behaved code was totally flaky! While touching the screen solidly, the code was reporting numerous make and break events that were not real. To make matters worse, I had made a lot of changes and due to a re-install of MPLAB-X, I had lost my backup history. Debugging did reveal real bugs in the logic which were soon fixed, but the phantom makes and breaks still persisted.

I eventually tracked it down to a puzzling observation. Most of the time, everything worked fine, but sometimes, the sense input pins were jammed in output mode with a value of "0". The results of my analysis was that the recommended fix for issue 62 actually causes a problem! I removed the errata #62 fix and the problems went away. What a relief; And that's why my blog post was delayed 😊 Some lessons (re)learned:

1. Errata recommendations can be wrong too.
2. Test often and test fully. Repeat old tests! Especially after unusual changes.
3. Use a version control system separate from your IDE and keep backups.
4. If it ain't broke, be really careful before you "fix" it!

Peter Camilleri (aka Squidly Jones)

SHARE THIS:



LIKE THIS:



Be the first to like this.

This entry was posted in [Embedded Development, Errata](#) by [Peter Camilleri](#). Bookmark the [permalink \[http://teuthida-technologies.com/?p=45\]](http://teuthida-technologies.com/?p=45) .