

Reality strikes

M. Simon



I started writing the below a few days ago, but I think it deserves a preface. The following report pains me to write. I dislike writing unrelieved darkness. In addition, the tools described are critical to some projects I had hoped to make a few bucks on. One is a low-volume teaching tool and one is a high-volume consumer appliance. So in a way, I am sinking my own ship. But the truth is more important than my own fortunes.

===

Life can be interesting - way more interesting than I like. I downloaded the driver for the [Adafruit programmer](#) [1] I bought. I installed the driver required and I got stuck. My PC doesn't seem to recognize the programmer. Since the "power good light" is controlled by a pin on the programmer [according to the schematic](#) [2], I plugged in the board I wanted to program ([MCU10 Developer](#) [3]) which has a power good LED connected directly to the 5 volt supply. Well that lit up. OK. Maybe I soldered the LED on the Adafruit programmer in backwards. Out with the trusty voltmeter. Zero volts across the LED. So even if I had soldered the LED in correctly, it is not going to light. Am I going to debug this mess? Maybe some day. But the whole Windows operating system has become so complicated that it might take weeks. I don't have weeks. In addition, the Adafruit Software suite requires you to program in C. I hate C. I will do it if some one pays me.

I architected an aerospace CAN bus tester in C - I designed the software system and handled the hardware issues. I did very little C programming. I did use Forth for hardware debugging. The CAN bus tester required considerable effort to work around the C compiler. Since I know computers from the inside, I was able to figure out what had to be done by looking at the assembler code the compiler turned out.

But I'm not going to use C for fun or in one of my own projects. I hate the stack thrash C CALLs cause. I hate being at the mercy of the C compiler. And besides, with 32 bytes of RAM, assembler or a non-resident Forth is way more appropriate.

I may get back to Adafruit. But I have an alternative. Local Atmel FAE, John Corral,

Reality strikes

Published on Electronic Component News (<http://www.ecnmag.com>)

sent me an [STK600](#) [4] programmer with an ATtiny10 programming adapter. So I'm going to break that out and give it a try.

Atmel keeps the data pages for the STK600 pretty well hidden. John Corral to the rescue again. Here is the [STK600 Technical Data](#) [5] reference page. Following the links there to the [ISP Programming](#) [6] page and then to the [In-System Programming of an External Target System](#) [7] page, I found out that the reset line can have a pull up of as low as 2.2K (I use 100K) and a capacitor as large as 10uF (I use .1uF) so I should be good to go. Let me add that after looking through the Studio 6 software, the same pages can be found in that suite. Look under "Tools". None of that "where to find it" information can be easily found at the Atmel www site.

The first thing to do is to plug in the USB cable and make sure the power switch on the upper right corner of the STK600 board is ON. I got a RED power ON light and two green lights when I turned the switch on. OK! Then I fired up [Studio 6](#) [8] and went to the tool page and selected the STK600. Then on to the Device page and yes - the ATtiny10 is selected. So far so good. Now back to the [In-System Programming of an External Target System](#) [7] page. That page says to use the ISP/PDI connector. If your eyes are as bad as mine, you have to use a magnifier to find pin one. When the cable is properly installed, the Red Stripe is to the right. (that assumes the USB cable is also to the right). Before connecting my [MCU10 Development board](#) [9], I went to the programming page of the STK600 (the icon with the lightning bolt through it), and I had to do a firmware upgrade first. It was quick. By the time I was done writing about it, it was finished.

I hit the lightning bolt again and chose the STK600. Without the target connected, I checked the device voltage. It was 5.0 volts - good to go. The maximum board current my MCU10 Developer draws with all the LEDs on is under 50 mA. (about 7 mA per LED plus a generous 15 mA for the ATtiny 10) the STK600 can source about 300 mA so away we go. I plugged the ribbon cable into my board and turned all the isolation switches on (not isolated) and read the signature.

And this is where things went bad wrong. I couldn't read the signature. I checked and double checked. Nothing. OK. The AVRISP mkII programmer had read the signature of this board last week using Studio 6. And after a few fumbles, it was reading it again. Back to the STK600 - still nothing. OK, maybe it was a faulty interface. Atmel FAE John Corral was kind enough to send me an ATtiny10 programming board with the STK600. So I put one of my ATtiny10 chips in the socket (use tweezers) and then mounted the board to the STK600. I got the same error. But the programmer did recognize the attached programming board ("card stack" they call it) - which indicates that the connections were likely good. So I emailed John and he suggested this [beta software upgrade](#) [10]. That tells me that things must be getting desperate if a beta upgrade is the best suggestion Atmel has. Well I'm getting desperate too. The beta upgrade didn't improve the situation. (note - this whole adventure upset me considerably so some of the events may not precisely conform to the exact time line - I did keep this narrative close though)

I'd like to close with part of an e-mail I sent John Corral:

Reality strikes

Published on Electronic Component News (<http://www.ecnmag.com>)

===

I would like to help you in your recovery plan (you do have one I hope).

1. I can make the Gerbers for my test board available to you if that will help. Please do not pass them around. It is a product if the AVRISP mkII can be brought up to snuff.
2. The AVRISP mkII needs to be brought into spec. I don't need a full up (with case) prototype to test. An uncased model (I will not divulge your secrets) or prototype will do. Do the layout quick (if it needs a new one) - the important thing is to get something you can test. And I can test.
3. The STK600 will need similar work - I suspect a software problem - but who knows.
4. What else can I do to help? Ask me.
5. Put a Tiger Team on this problem. Your credibility is at stake.

I have done quite a few recovery plans in aerospace. I was quite good at it and always recovered well within deadlines and budgets. (budgets are not hard - but coming in well under deadline is) Draw on me as a resource. No charge unless you think my contributions are of value. And you determine the value. This is not in any way a contract. I want to help you be a success with the ATtiny10 family. Nice parts if they work according to spec.

Let me add that you have been a prince in your efforts to support me. I hope Atmel takes that into account when your next review comes up. (feel free to pass this on to corporate)

===

And that is where things stand today.

M. Simon's e-mail can be found on the sidebar at [Space-Time Productions](#) [11].

Source URL (retrieved on 04/28/2015 - 12:43am):

<http://www.ecnmag.com/blogs/2012/08/reality-strikes>

Links:

- [1] <http://www.adafruit.com/products/46>
- [2] <http://ladyada.net/images/usbtinyisp/usbtinyisp2sch.png>
- [3] <http://www.ecnmag.com/blogs/2012/08/i-found-bug>
- [4] <http://www.atmel.com/tools/stk600.aspx>
- [5] <http://www.atmel.no/webdoc/stk600/stk600.html>
- [6] http://www.atmel.no/webdoc/stk600/stk600.programming_isp.html
- [7] http://www.atmel.no/webdoc/stk600/stk600.programming_ext.html
- [8] http://www.atmel.com/Microsite/atmel_studio6/
- [9] <http://spacetimepro.blogspot.com/2012/07/atmel-attiny10-development->

Reality strikes

Published on Electronic Component News (<http://www.ecnmag.com>)

board.html

[10] http://www.atmel.no/beta_ware/

[11] <http://spacetimepro.blogspot.com/>