Computing stuff tied to the physical world

- Intro|
- Docs|
- Forum|
- Book|
- Shop|
- About|
- Archives|
- RSS Feed

AC Mains

More sensitivity

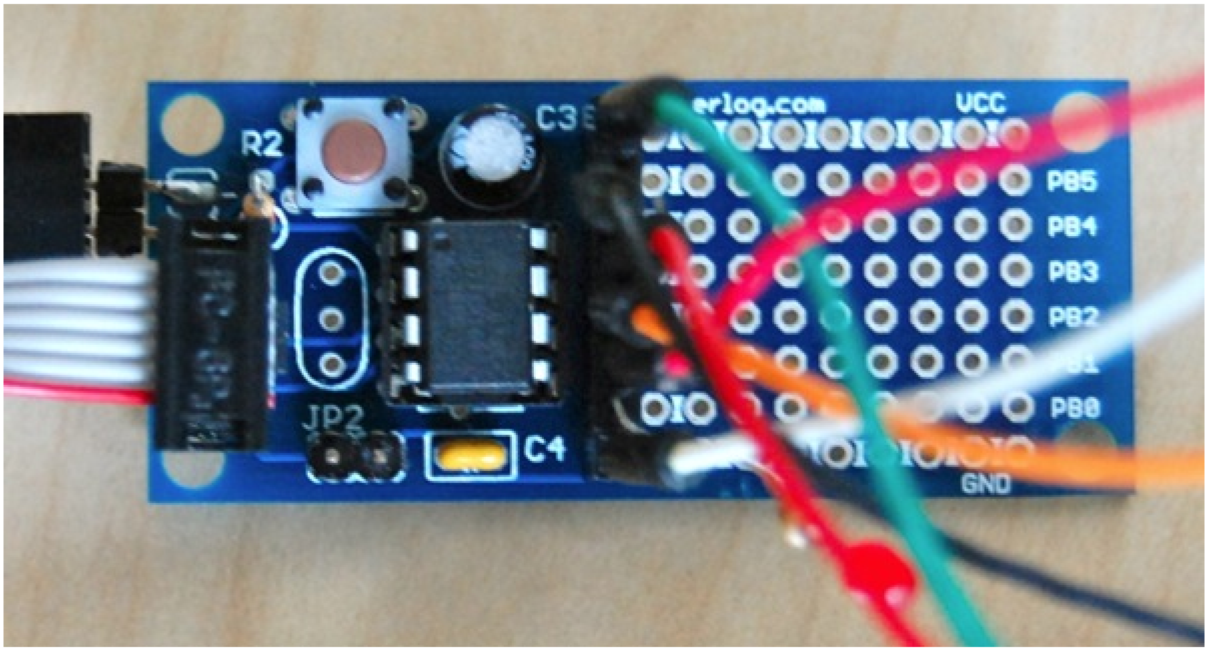
In Hardware on **Oct 9, 2011** at **00:01**

Yesterday's post was nice, but it stepped over one teeny weeny little detail: the ATmega's ADC isn't sensitive enough to measure AC signals in the millivolts range. With a 3.3V reference, each step is ≈ 3.3 mV, while the signals at low power levels are close to a single ADC step. I could bump the sensitivity slightly by using the 1.1V bandgap as ADC reference voltage, but that still only gets me to 1.1 mV steps. Hardly enough to apply filtering to and set thresholds.

Ah, but there's a way out… (there always is!)

The ATtiny85 is the smaller brother of the ATmega with a frighteningly low pin count of 8. The key is that it has a *differential* ADC option, i.e. it's able to measure the voltage between two points (within 0..VCC) *and* that it has an optional 20 x analog signal amplifier built-in, when used in differential mode. As reference voltage, there is 1.1V, 2.56V, and VCC – providing a couple of ways to tweak the actual range and sensitivity.

Since I had the right ATtiny85 lying around, as well as TinkerLog's prototyping board, I thought I'd give it a go:



The problem with an 8-pin chip, of course, is that it only has 5 I/O pins (6 if you're willing to use the RESET line and lose the ISP programming mode – which I wasn't). That's not much to interface with. The ATtiny85 I used has 8 Kb flash and 512 bytes of RAM, so that in itself should be sufficient for all the code I need.

There's no boot loader. There's no ready-to-use serial port. And it's not an Arduino-compatible system, so you can't just write sketches for it, right?

Not so fast. First of all, there's the LCD Plug which needs only 2 I/O lines. That leaves me 3 pins: 2 for the differential analog input, and 1 for a debug LED. *Plenty!*

The LCD needs the Ports library. Which needs the Arduino IDE and runtime. Ah, but there's more good news: there's an arduino-tiny project which defines some boards and a set of "core" runtime source files which are intended to run on ATtiny85's and other non-mega AVR chips. That's quite a feat, btw, once you consider just *how* limited an ATtiny85 is (ASCII art by the arduino-tiny project):

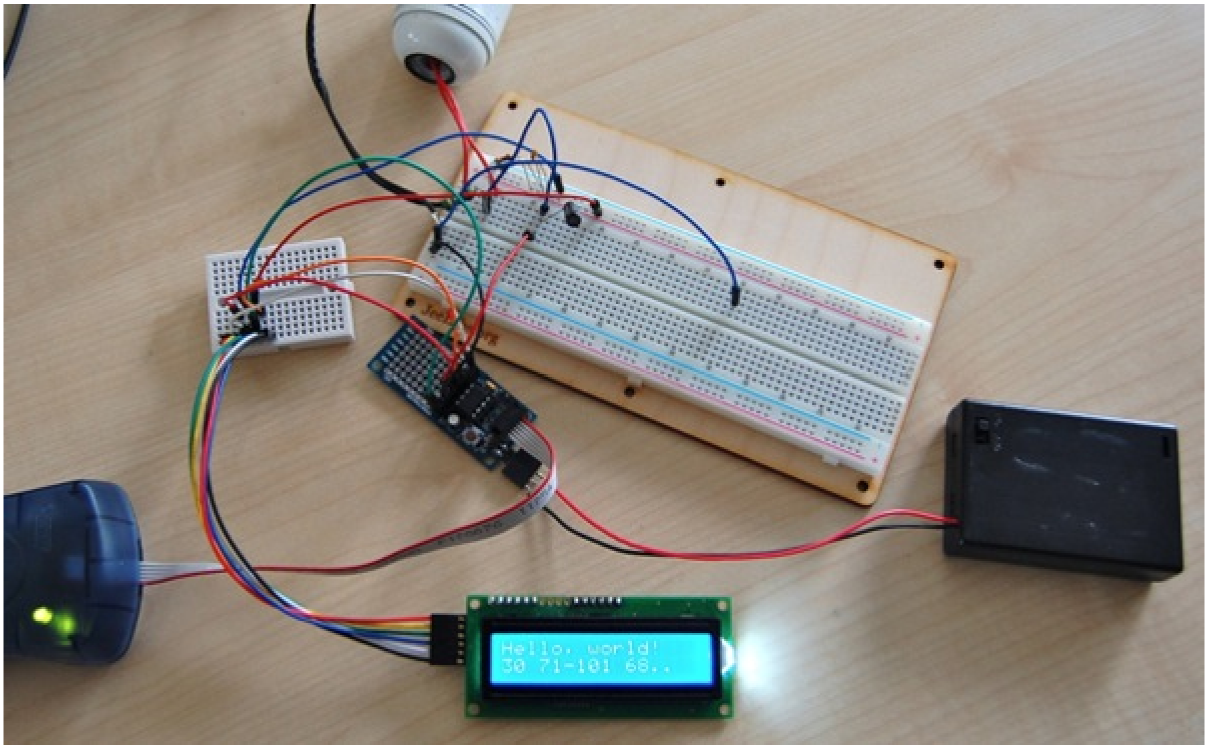
```
                             +-\/-+
    Ain0       (D  5)  PB5  1|    |8   VCC
    Ain3       (D  3)  PB3  2|    |7   PB2   (D  2)  INT0  Ain1
    Ain2       (D  4)  PB4  3|    |6   PB1   (D  1)        pwm1
                       GND  4|    |5   PB0   (D  0)        pwm0
                             +----+
```

Still, they managed to support a couple of digital and analog I/O pins, with support for the millis() timer, analog in and out, and more. With this installed, I can write sketches and upload them via the AVR-ISP. *Pretty amazing!*

Sooo… just had to make a few small and obvious tweaks, and the Ports library works. There's exactly *one* port (AIO1 = PB2, DIO1 = PB0, IRQ = PB1). Note that I'm using the new Arduino IDE 1.0 (beta).

That leaves me a whopping two pins for the differential analog input, which is what this is all about, after all.

Here's my setup (hooked up to the safe 20 VAC brick):



I was too excited to tidy up once the LCD hookup worked. It would all have fitted on a single (mini) breadboard.

Tomorrow, I’ll show you how this crazy little setup measures up…

▶ View 3 Comments
« Before Moving averages Oct 8, 2011
AfterAC current detection works! Oct 10, 2011 »

Recent Posts

- Turning the page on 2015
- Tying up 2015’s loose ends
- Switching to a new server
- A diversion into FPGAs
- A fingernail vs a moon lander
- Pi-based STM32F103 development
- Programmer PCB Triple Play
- Talking to an STM32
- The world of STM32
- Making a sharp turn
- IDE w/ LPC824, part 2
- Arduino IDE w/ LPC824
- Meet the Tinker Pico (again)
- Getting back in the groove
- Shedding weight
- Bandwagons and islands
- Hundertwasser
- Space tools
- Clojure and ClojureScript
- A feel for numbers
- Lessons from history
- Doodling with decoders

- Could a coin cell be enough?
- Forth on a DIP
- RFM69s, OOK, and antennas
- Greasing the “make” cycle on Mac
- Low-power mode :)
- FTDI over WiFi: esp-bridge
- Moving up a (few) level(s)
- Code for the LPC8xx
- Introducing the LPC824
- RFM69 on ATmega
- RFM69 on Raspberry Pi
- Micro Power Snitch, success!
- Micro Power Snitch, part 7
- Micro Power Snitch, part 6
- We interrupt this program
- Dataflash via SPI
- Analog on the cheap
- Emulating EEPROM
- From LPC810 to LPC812
- Micro Power Snitch, last part
- Micro Power Snitch, part 4
- Micro Power Snitch, part 3
- Micro Power Snitch, part 2
- Micro Power Snitch
- Uploading over serial
- Bits, pointers, and interrupts
- LPC810 meets RFM69, part 3
- Volta makes the world go round
- Embedded Linux
- LPC810 meets RFM69, part 2
- LPC810 meets RFM69
- Eye Squared See
- Schematics and layouts
- Dip into the LPC810
- Garage Parking Aid
- Getting started, final episode
- Getting started, episode 3
- Getting started, episode 2

Archives

- December 2015
- November 2015
- October 2015
- September 2015
- August 2015
- July 2015
- June 2015
- May 2015
- April 2015
- March 2015
- February 2015
- January 2015

- December 2014
- November 2014
- October 2014
- October 2013
- September 2013
- July 2013
- June 2013
- May 2013
- April 2013
- March 2013
- February 2013
- January 2013
- December 2012
- November 2012
- October 2012
- September 2012
- July 2012
- June 2012
- May 2012
- April 2012
- March 2012
- February 2012
- January 2012
- December 2011
- November 2011
- October 2011
- September 2011
- June 2011
- May 2011
- March 2011
- February 2011
- January 2011
- December 2010
- November 2010
- October 2010
- September 2010
- August 2010
- July 2010
- June 2010
- May 2010
- April 2010
- March 2010
- February 2010
- January 2010
- December 2009
- November 2009
- October 2009
- September 2009
- August 2009
- July 2009
- June 2009
- May 2009
- April 2009
- March 2009

- February 2009
- January 2009
- December 2008
- November 2008
- October 2008
- ARM
- AVR
- Book
- Hardware
- Linux
- Musings
- News

- ### About

  "Jee" stands for JC's Environmental Electronics. This blog is maintained by Jean-Claude Wippler.
- Search for:

  Search
- **Alphabetical** and **Chronological** index of weblog posts.
- RSS Feed

- ### Tags

  AC Mains Analog Arduino Audio CNC DIJN Displays Electronics Events Flashback Graphics Heating Home automation HouseMon ISP JavaScript JeeBot JeeDay JeeHub JeeLink JeeMon JeeRev JeeSMD Linux Metering Musings MuxShield Network Oscilloscope POF Power RBBB Reflow RTOS Sensors Solar Teardown Toolkit What-If X10

Theme: **DePo Masthead** by Derek Powazek