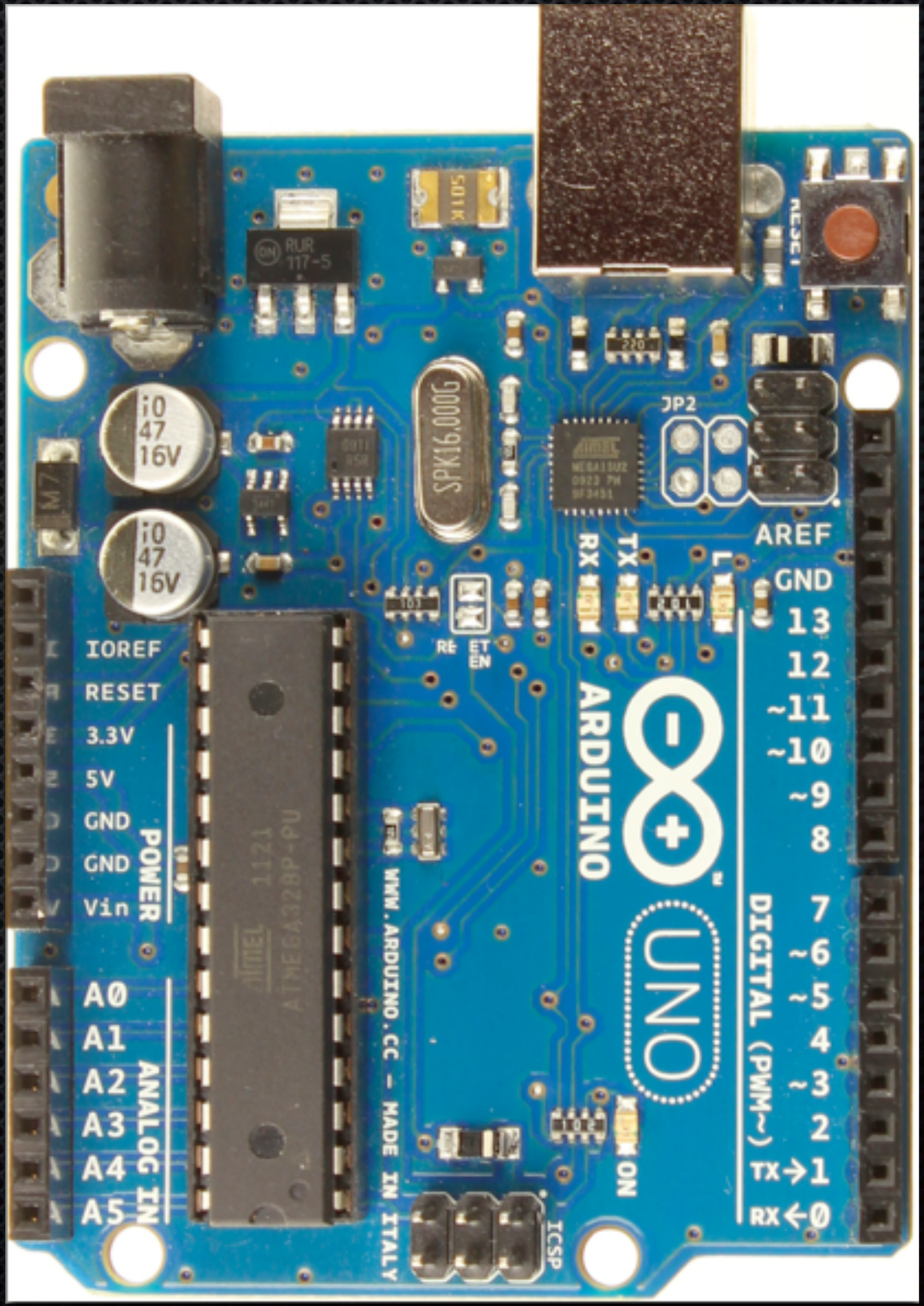
Beyond Blinking Lights: Real-world Lab Solutions Using Arduinos

Eric Ayars

California State University, Chico

Outline

- When you *should* use an Arduino
- When you *shouldn't* use an Arduino
- When you shouldn't use an *Arduino*

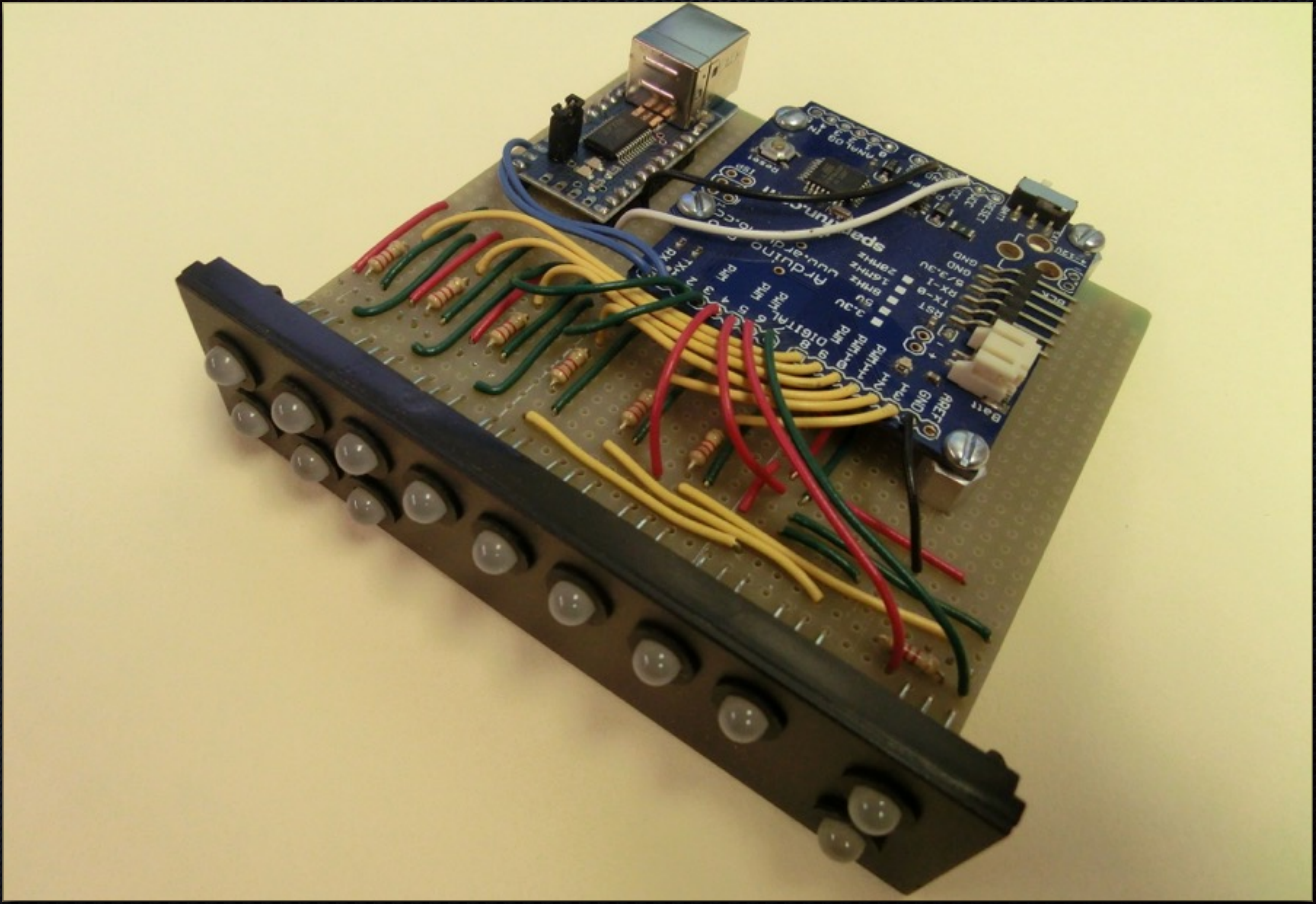


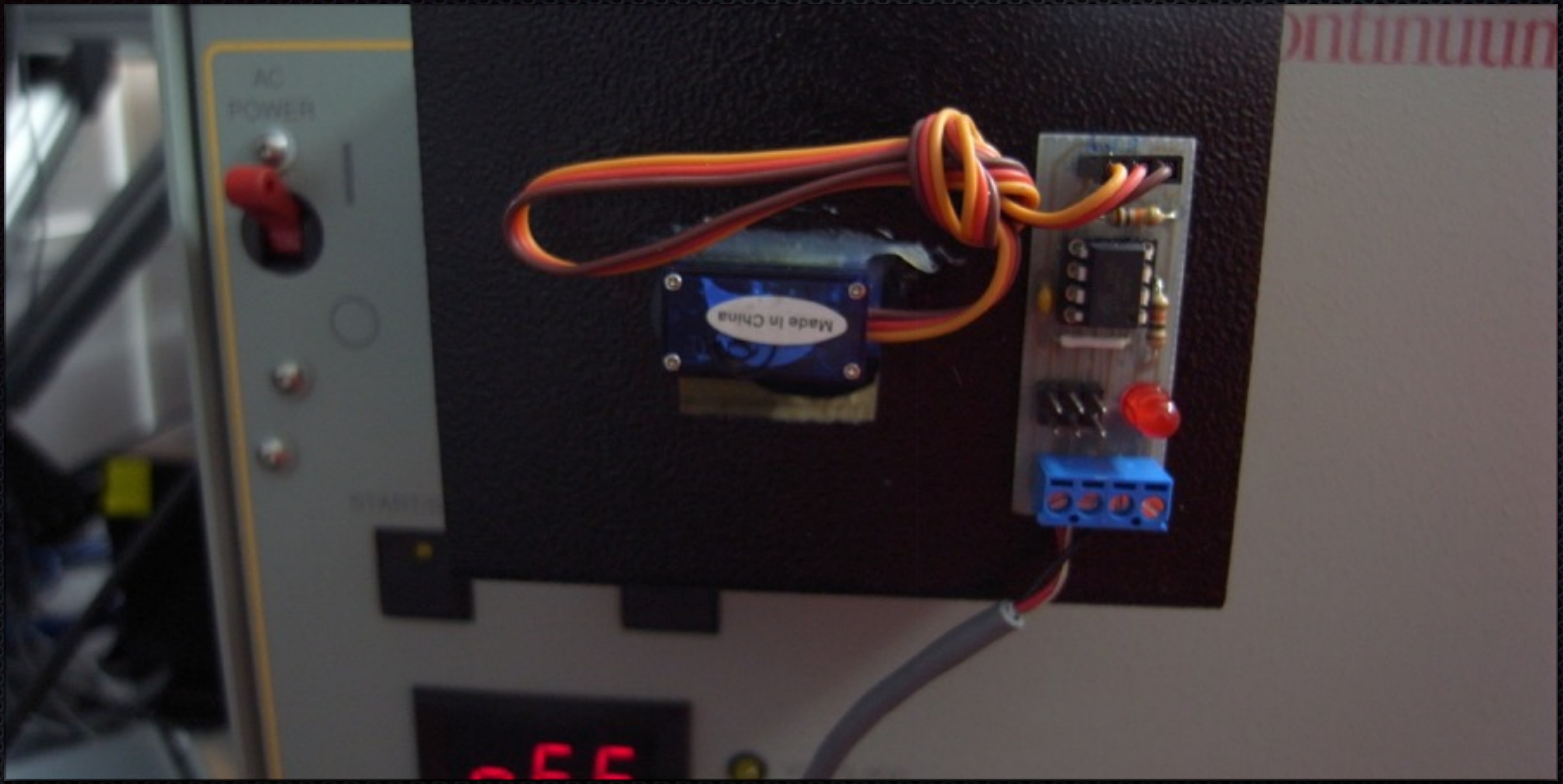
Microcontrollers are best at *single* tasks.

- Sub-instruments — doing *part* of a job very well.
- Communications bridging — talking to other chips or sensors
- Faking Data — generating test data/ signals
- New Instruments — Borderline case…

Not physics lab...

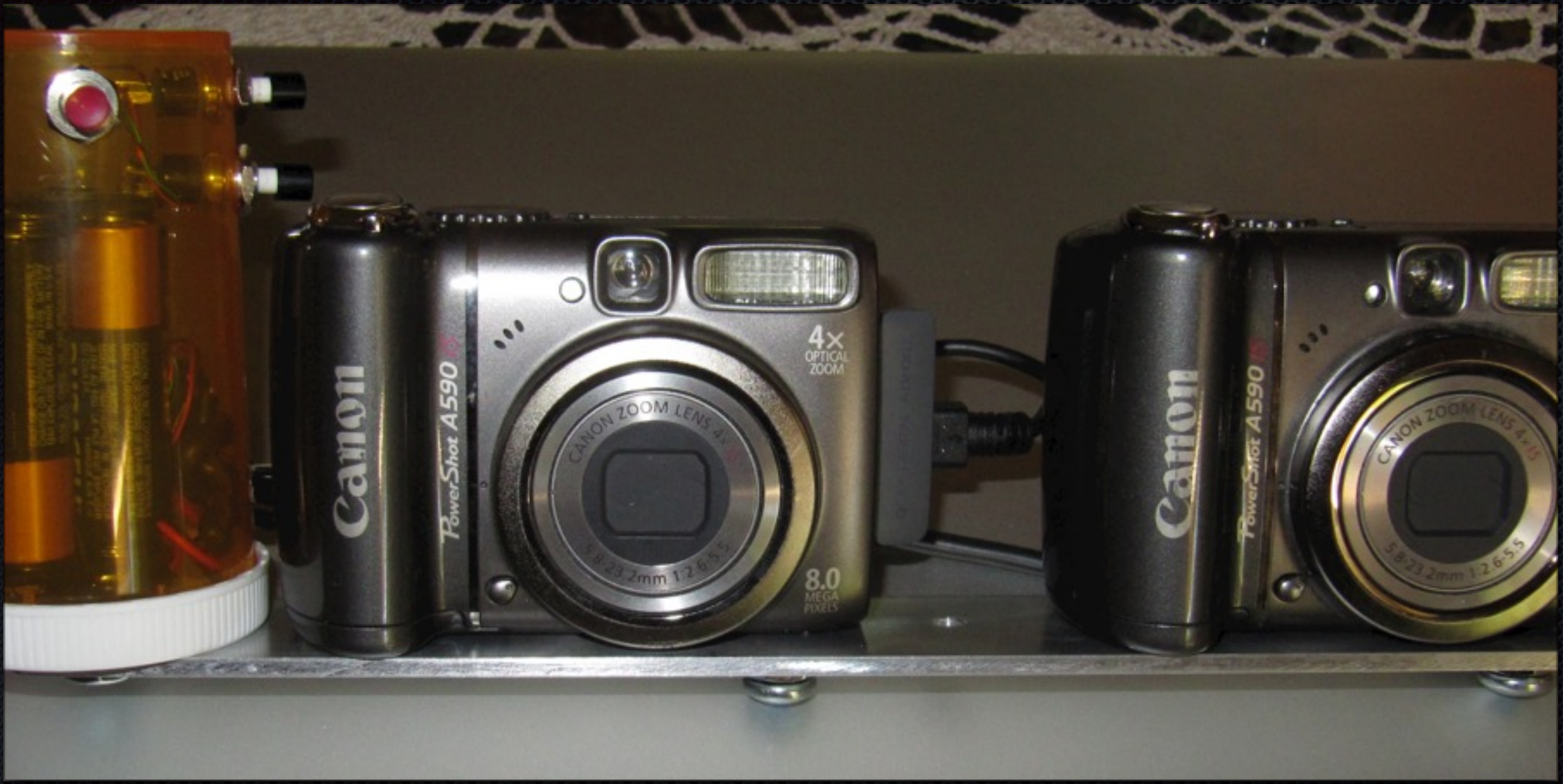
... but applicable to lab. Warning indicators, fail-safes, interlocks, etc.


Arduino
DIGITAL
PWM
GND
AREF
Batt



Sub-instruments

Here an “Arduino” controls a physical key-switch.



Part of the equipment

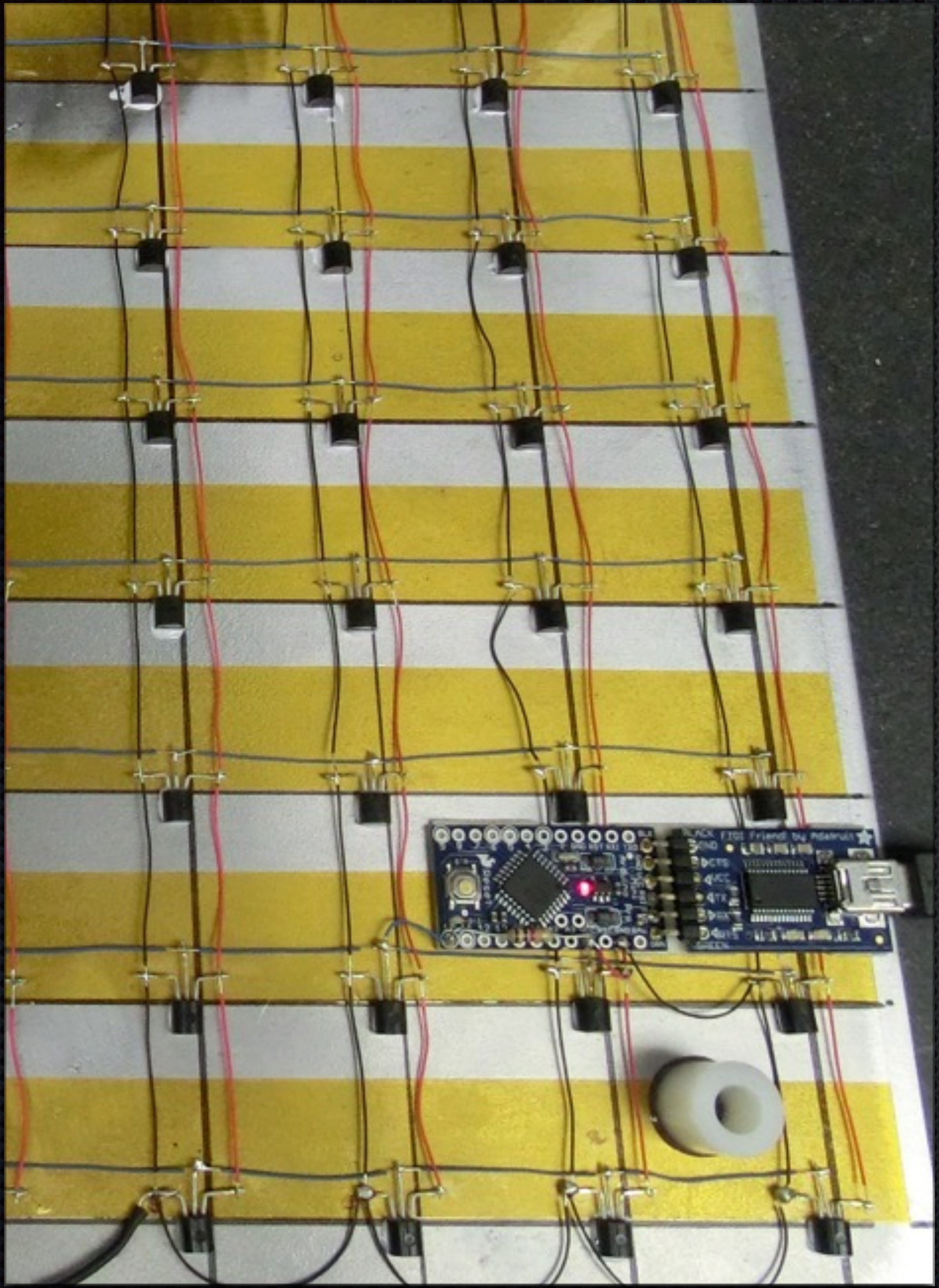
Here both cameras are controlled (zoom and shutter) with one set of buttons

BALDWIN LOCOMOTIVE WORKS
PHILADELPHIA
12

BALDWIN LOCOMOTIVE WORKS
PHILADELPHIA
12

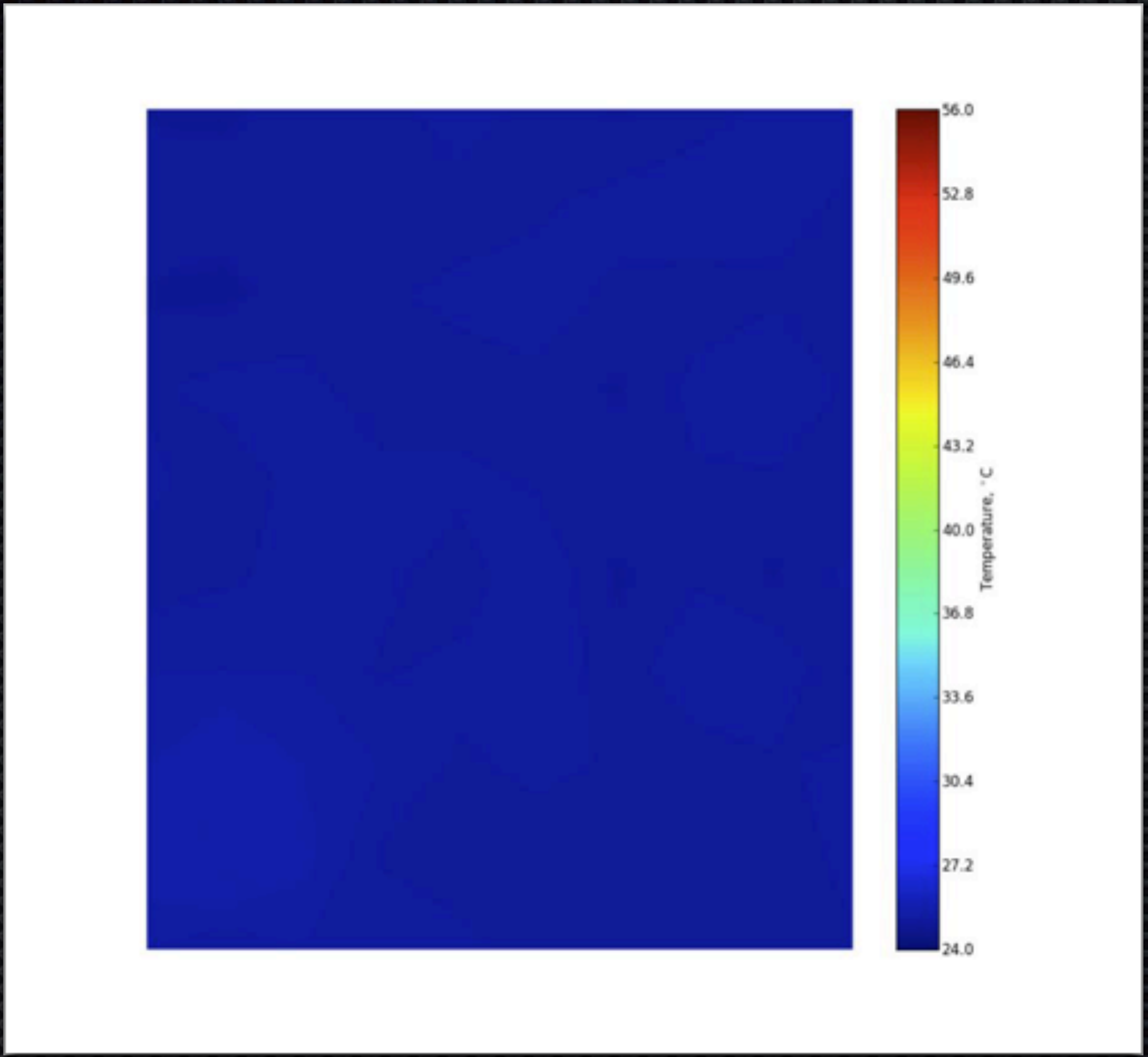
Other sub-instrument ideas

- Temperature controller
- Other PID control
- Position control
- Stepper motor speed/acceleration control



Sensor Interfacing

Measure things you
can’t measure
otherwise

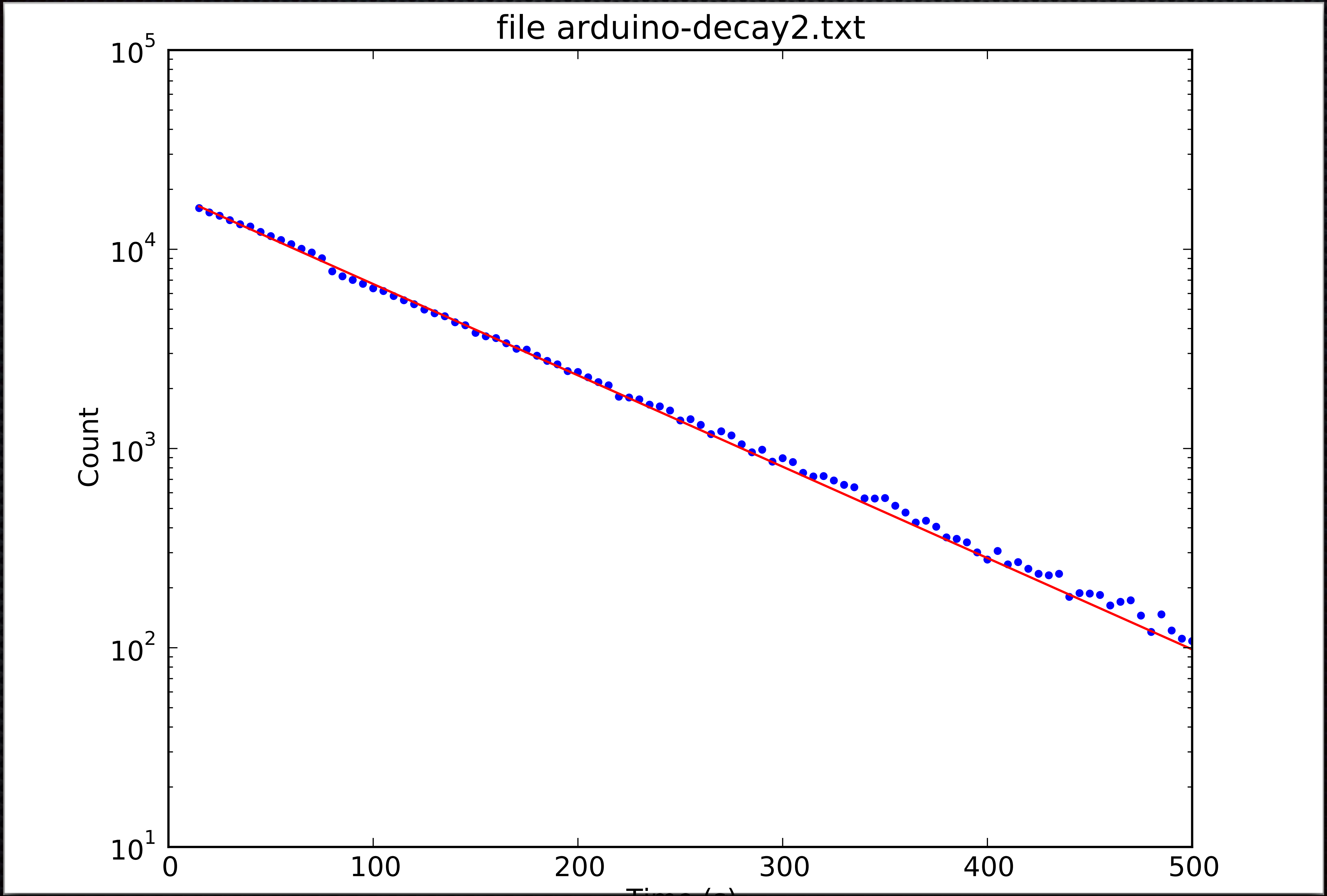


Temperature of a 30cm-square 5mm aluminum plate. Resolution: 0.1°C thermal, 3cm spatial. (Video sped up by a factor of 2.)

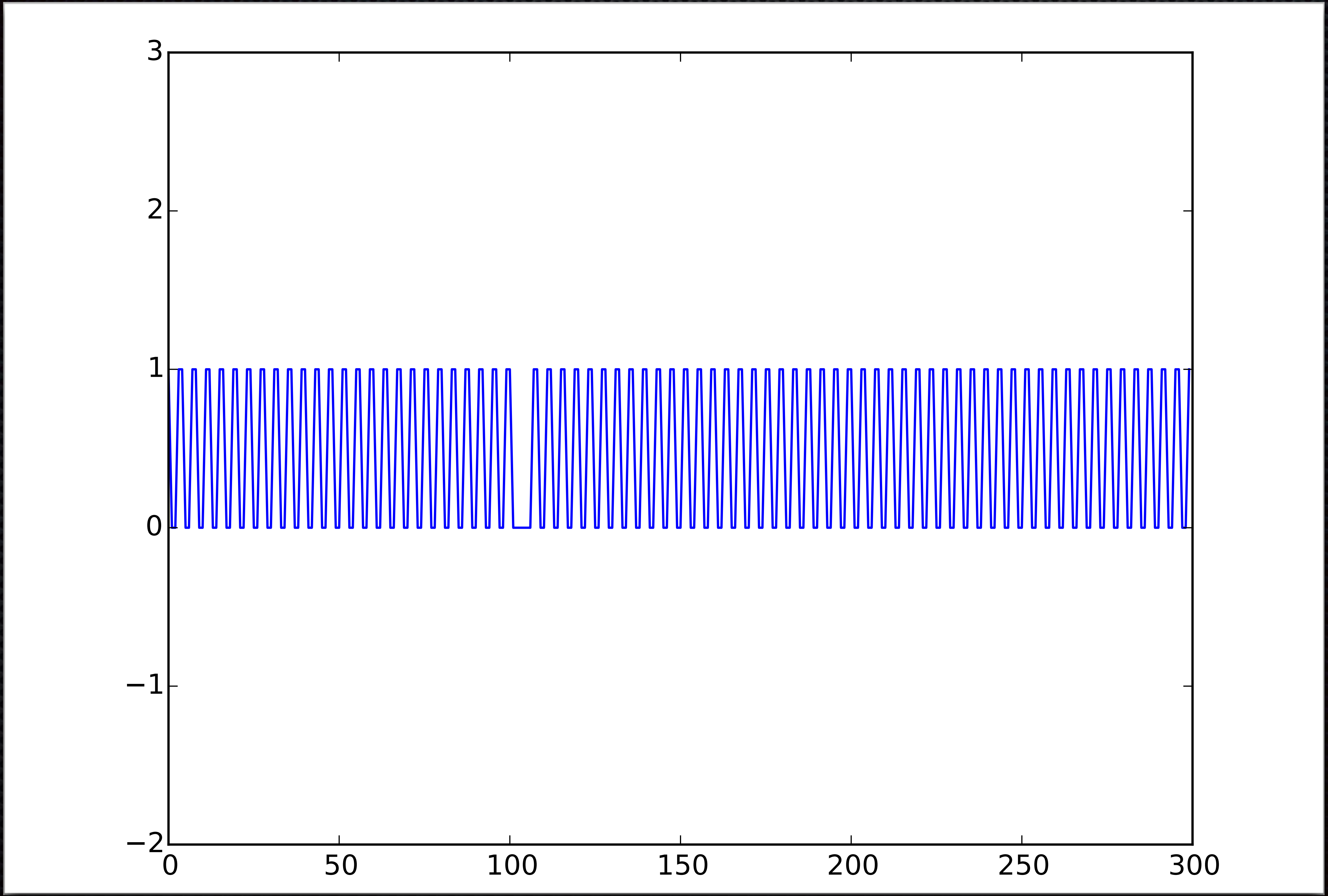
Communications Bridges

- I^2C: A/D and D/A converters Accelerometers, Gyros, Clocks, Thermometers, Barometers...
- SPI: Canon AF Lenses, SD Cards, PASCO sensors...
- One-wire: Thermometers, other environmental sensors...
- Serial:
  Old equipment

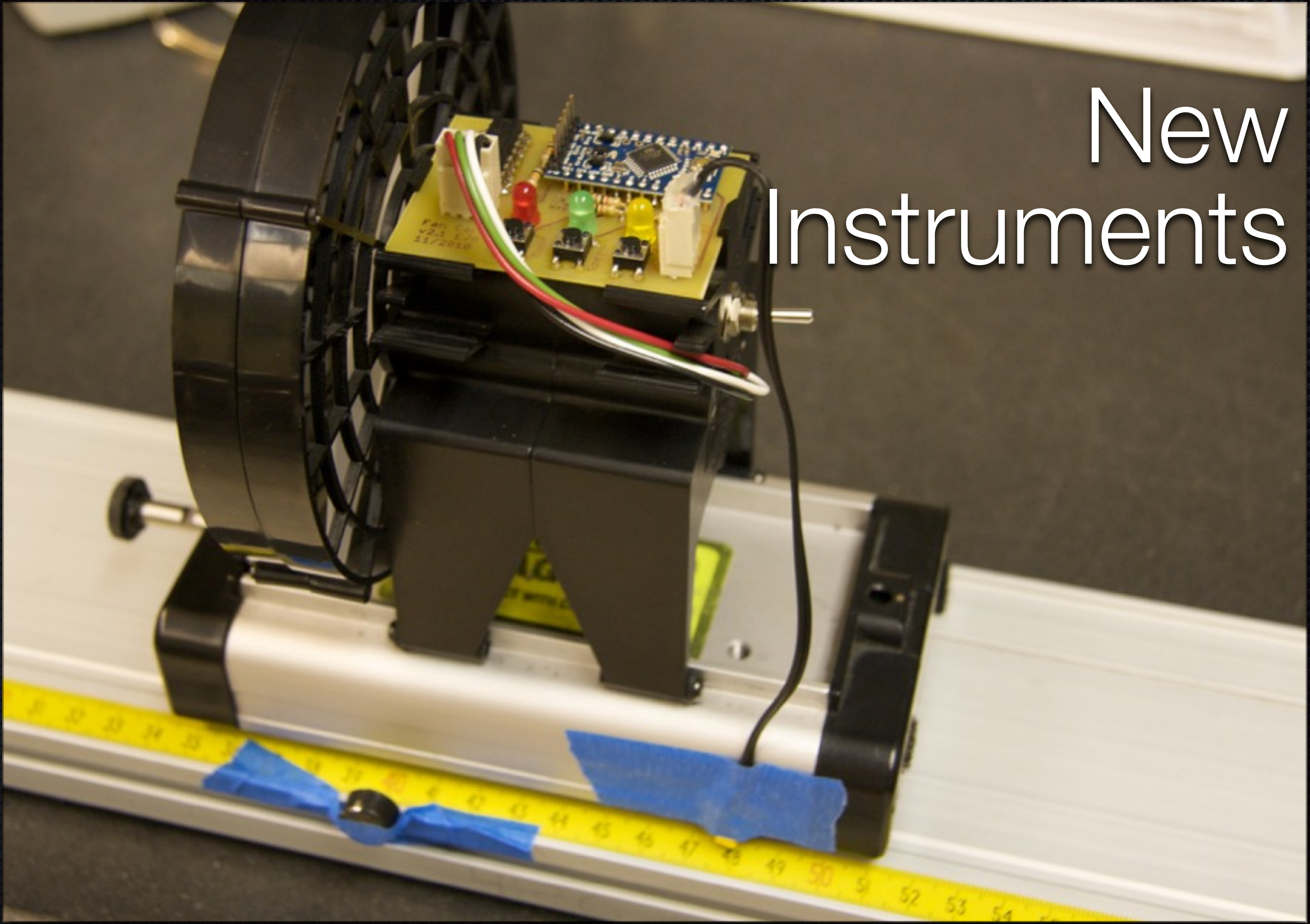
“Faking” data


file arduino-decay2.txt
Count
10^5
10^4
10^3
10^2
10^1
0
100
200
300
400
500
Time (s)

“Faking” data

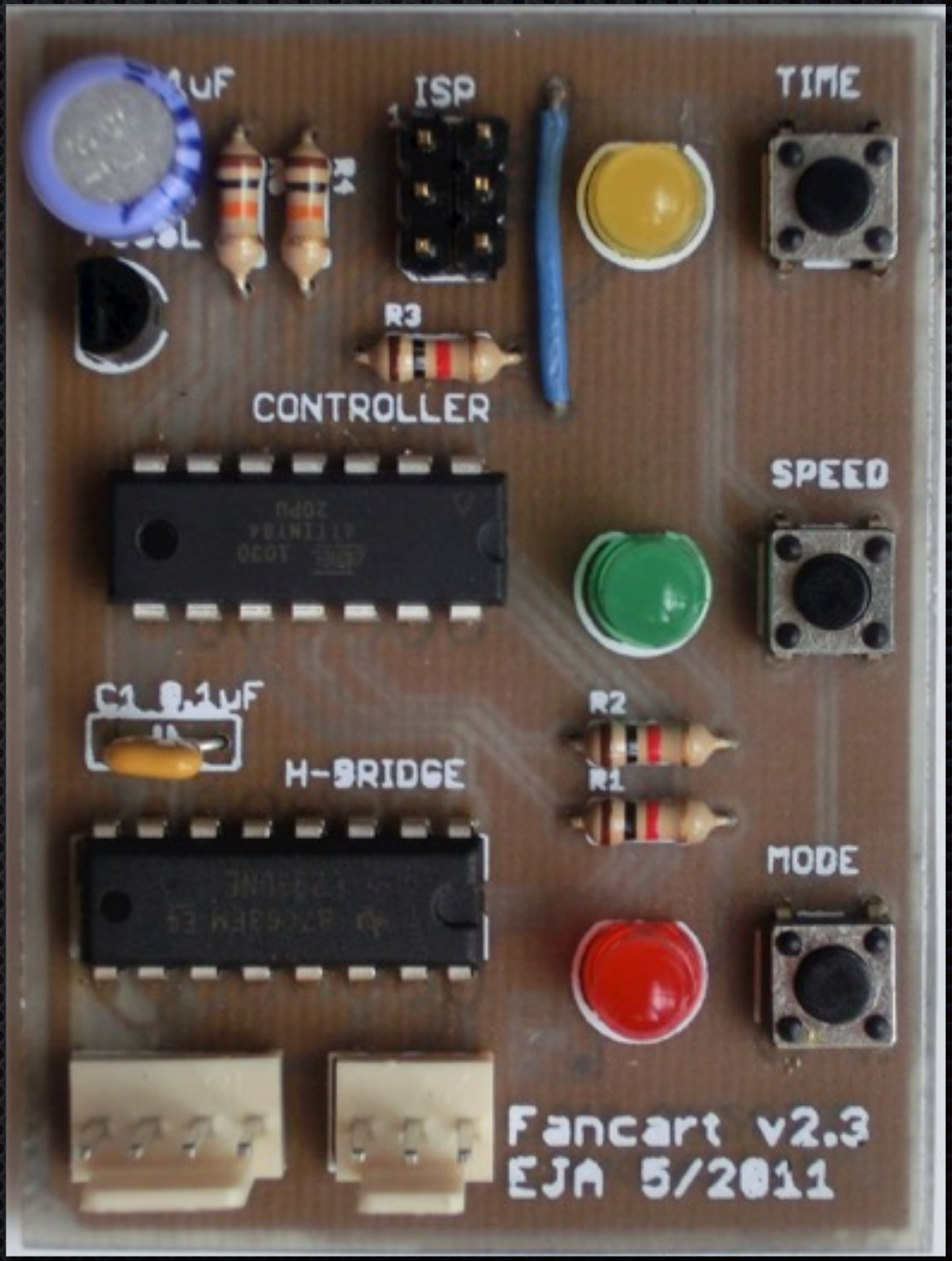


New Instruments



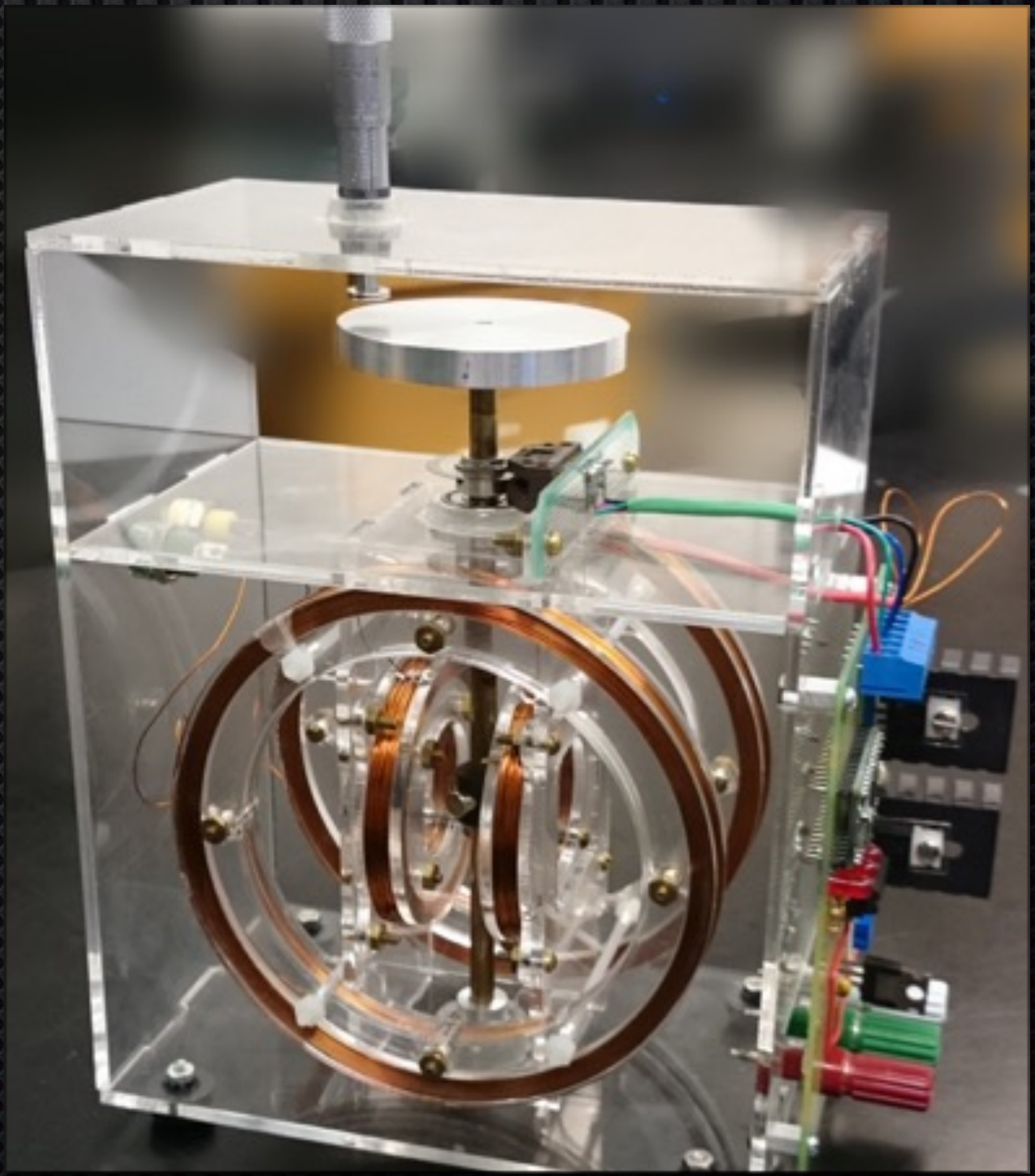
Flexibility in instrumentation

If you can do it with a small C++ program, you can (probably) do it with a microcontroller. And you can change the program.

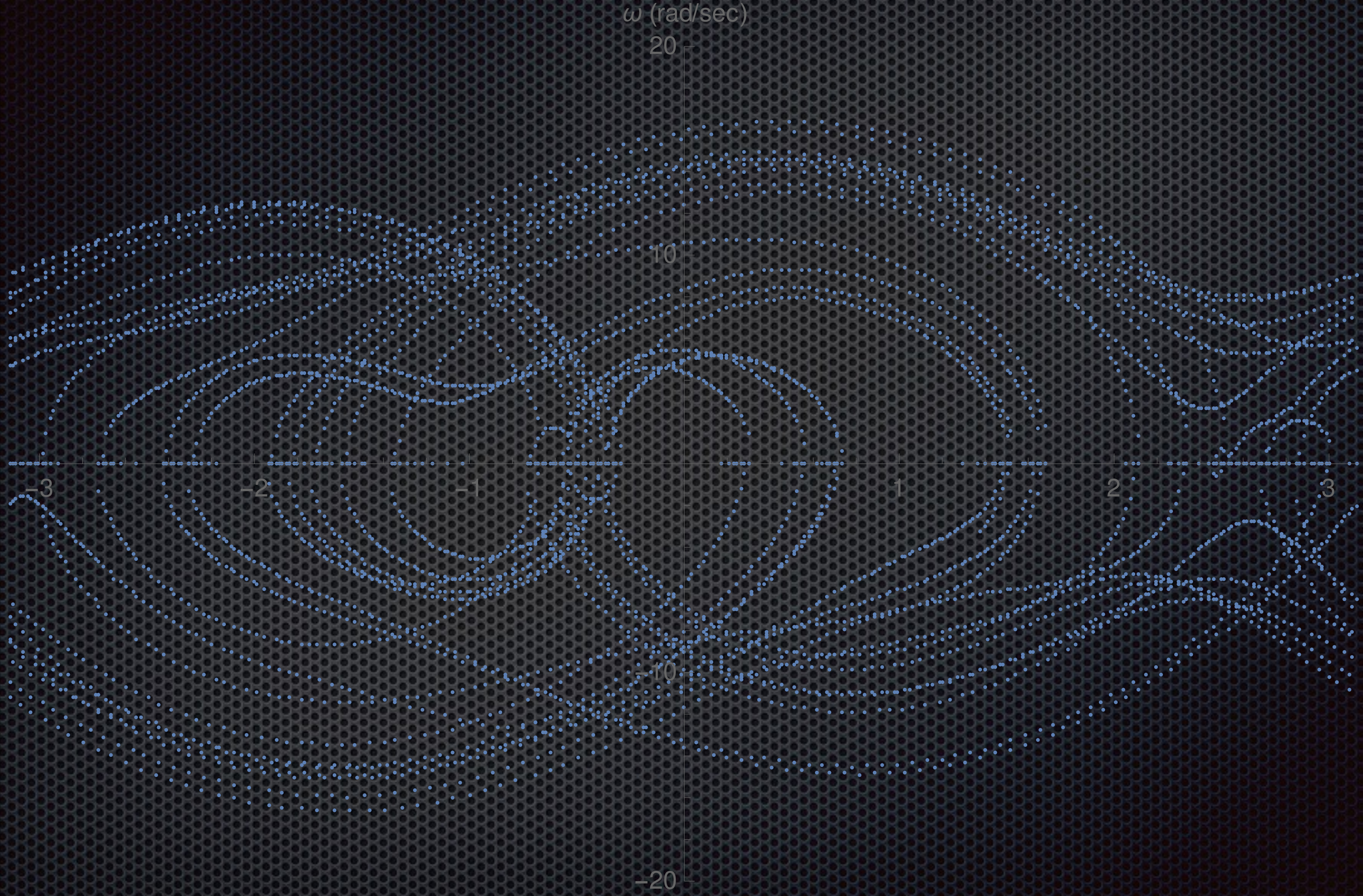


Mechanical Chaotic Oscillator

- Rotating dipole in oscillating B field
- All parameters are controlled by the microcontroller.
- SCPI commands through USB allow any computer to interface to this instrument.

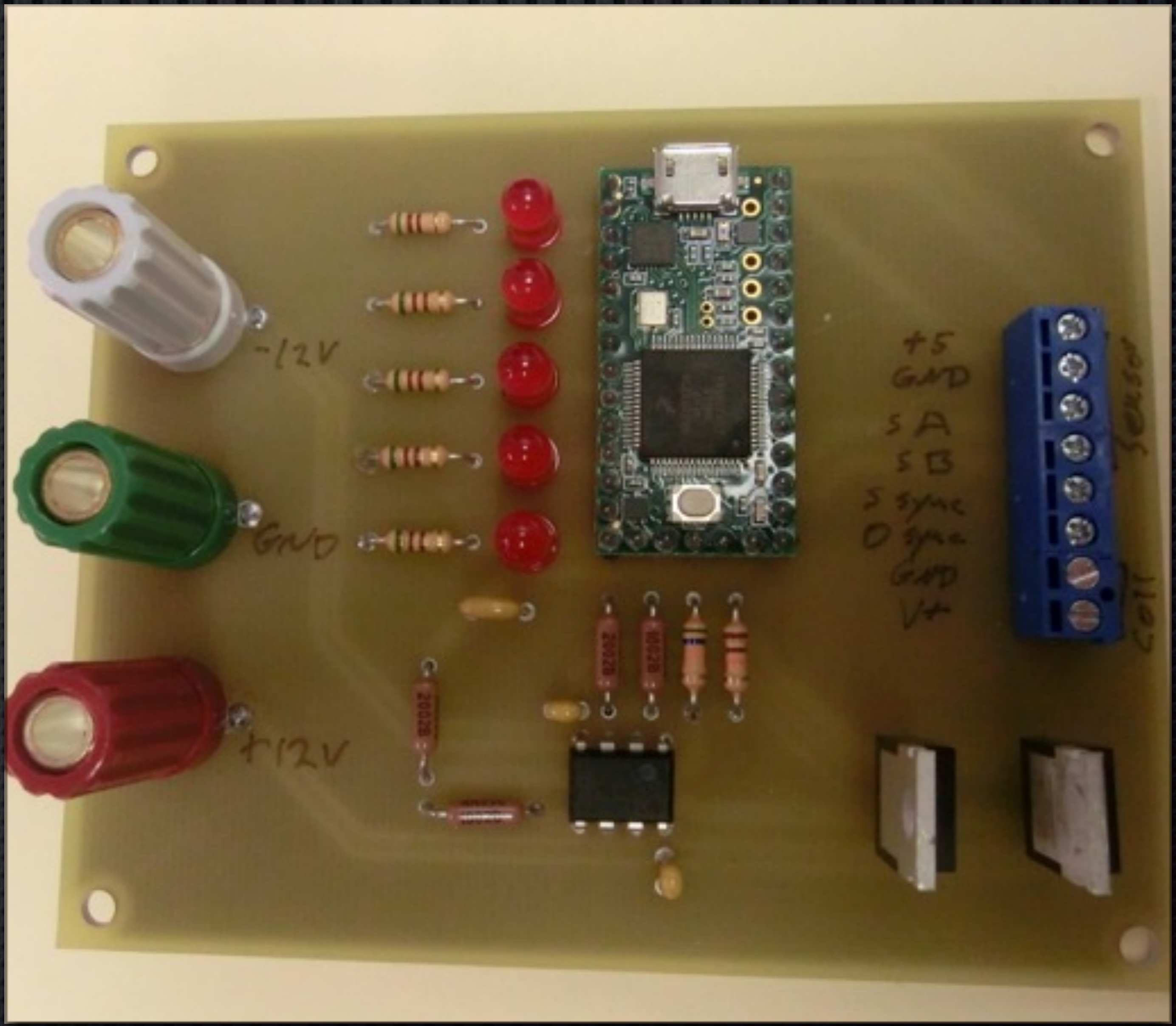


Amplitude 0.82



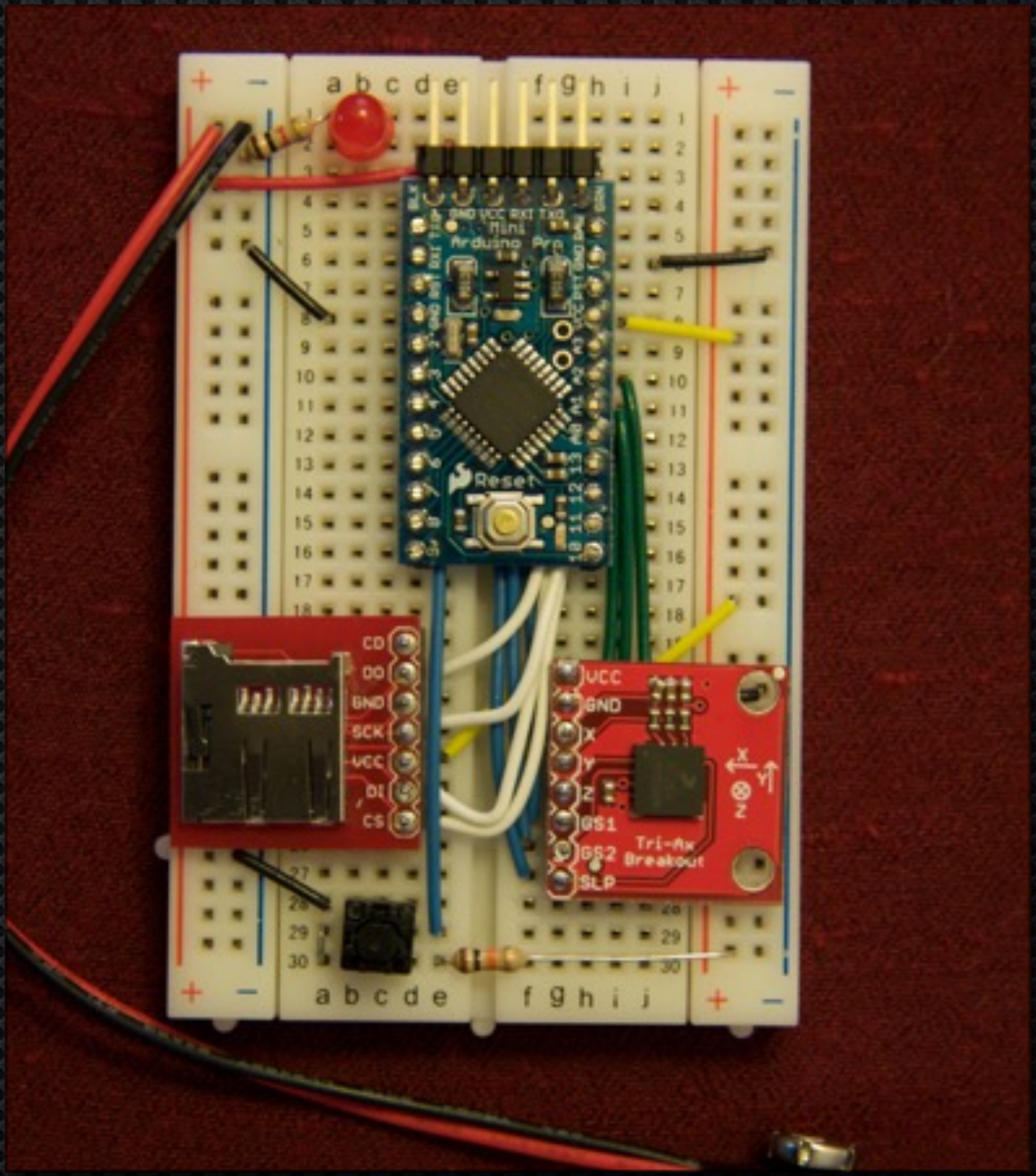
Flexibility, again:

- Low-Frequency programmable semi-intelligent sensor-equipped USB-interfaced SCPI-capable high-current arbitrary function generator



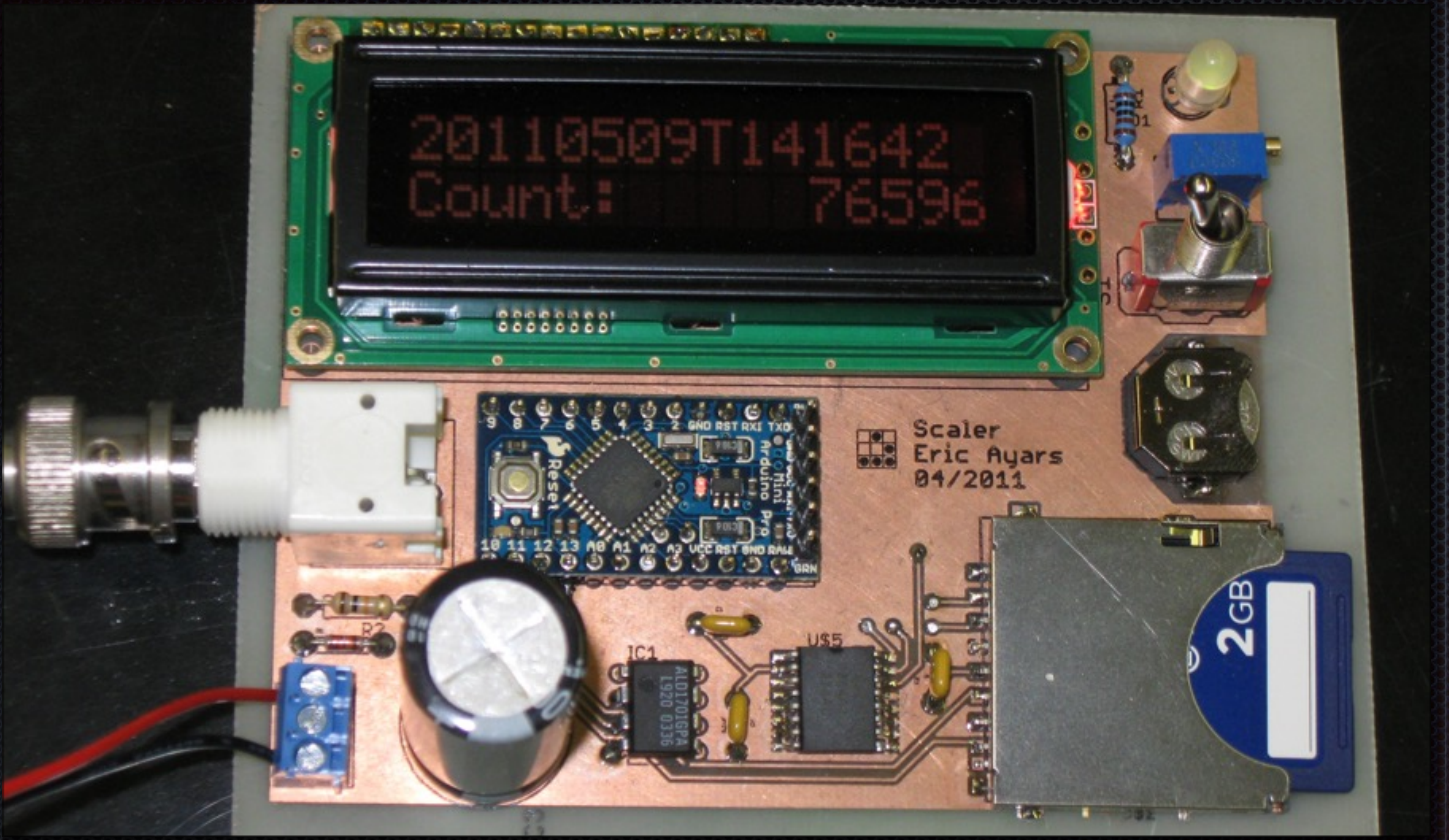
Datalogging

Collect and save data
for later analysis

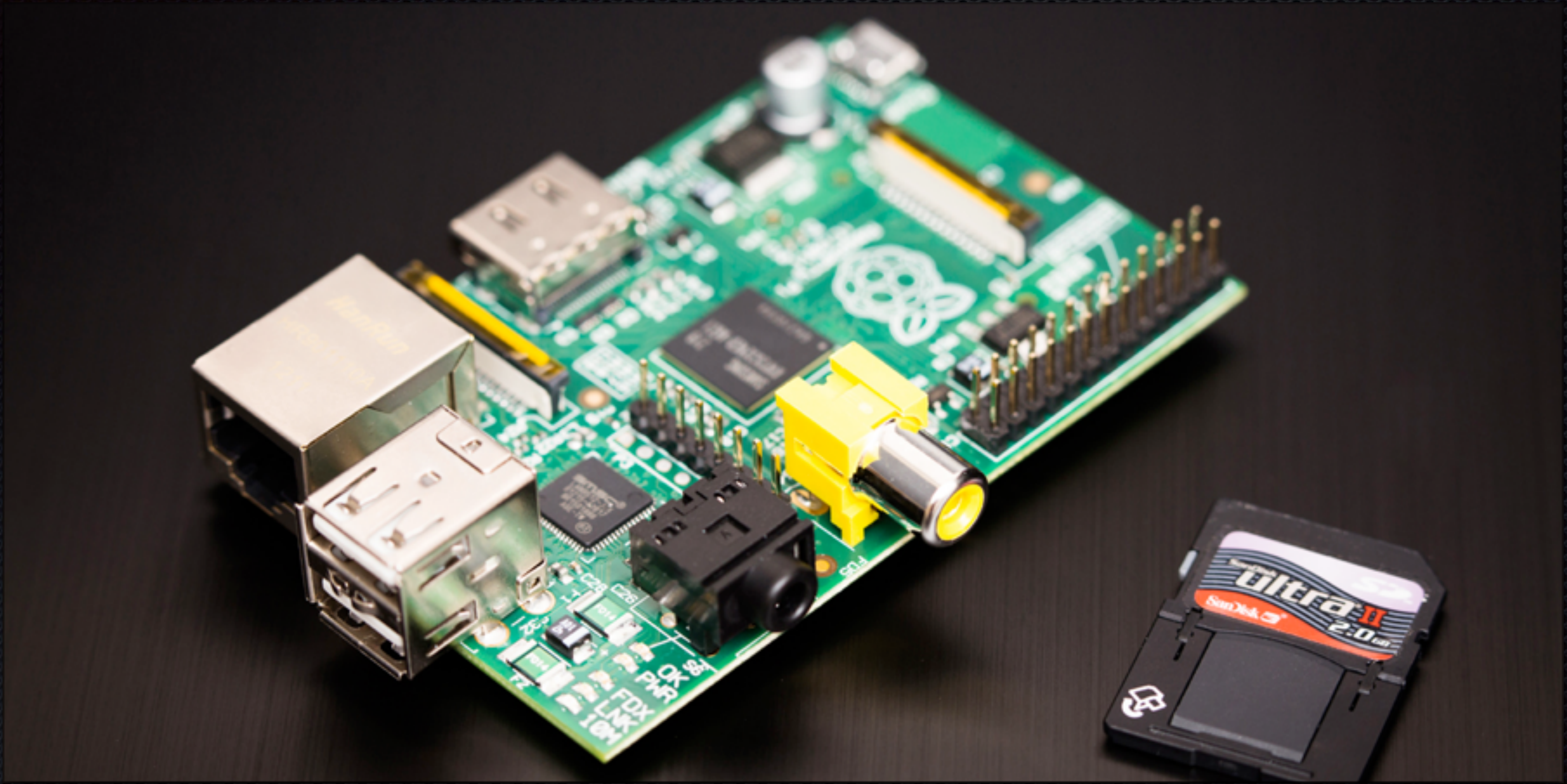


Arduino is only “fair” as a datalogger.

- Speed: 10 Hz for saving data to text file on SD card
- Precision: 10-bit A/D conversion
- If you don’t need speed or precision, though, it’s still useful.
- …and you can add external A/D converters for better precision.



This is probably a bad idea.

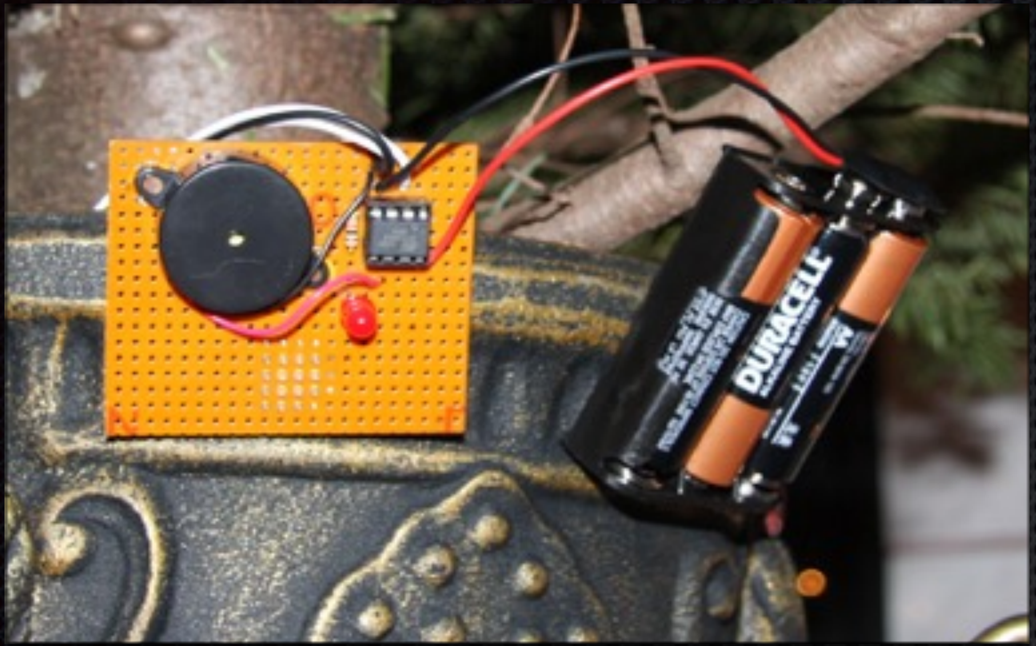


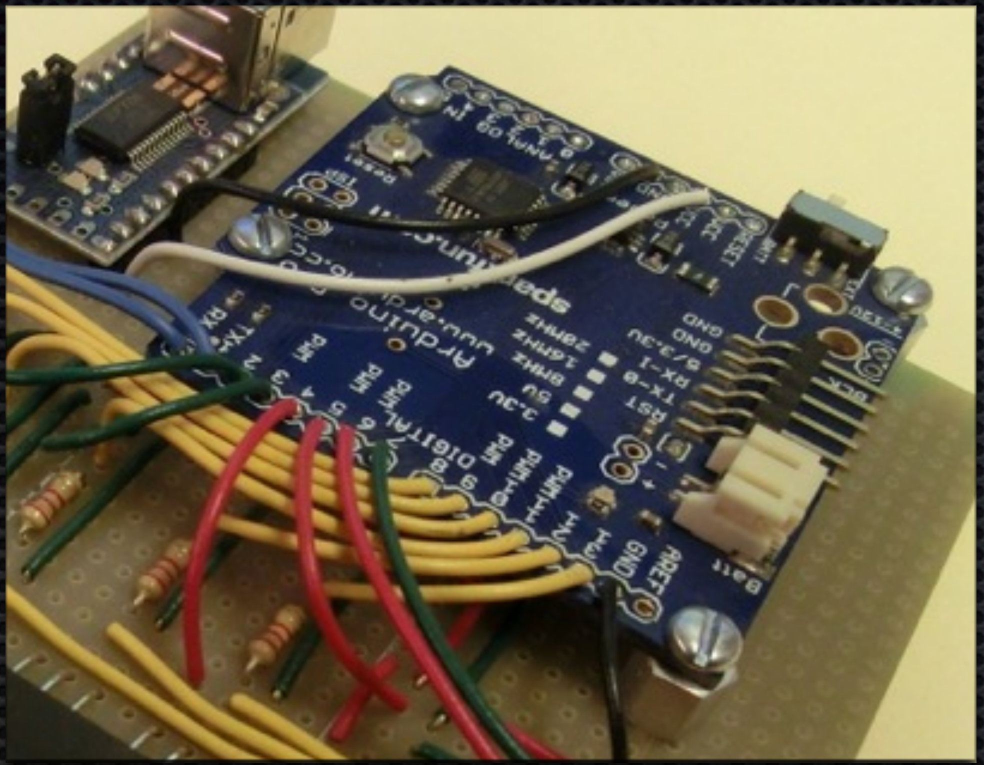
Here’s why it’s a bad idea.

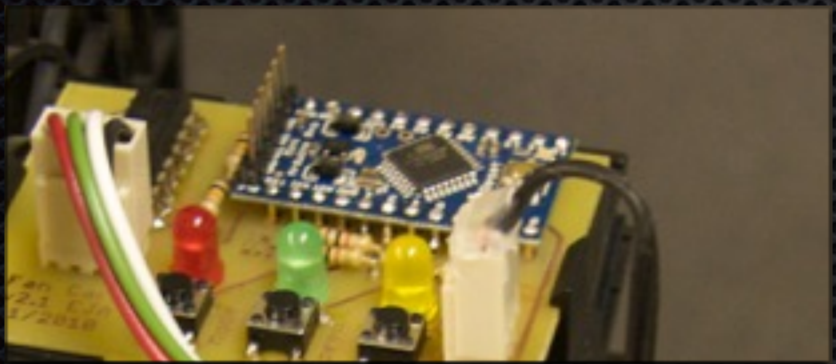
If you find yourself building a computer, just buy a $35 computer instead.

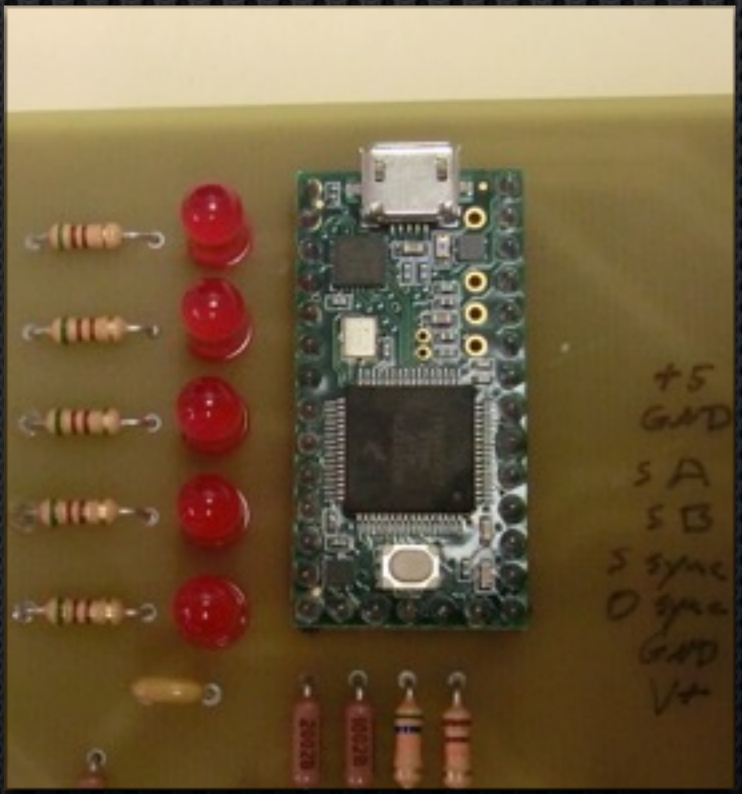
When to not use an *Arduino*

- Do you need an Arduino’s complexity, or will a single chip do the job?
- Do you want breadboard capability?
- Do you need *more* capability than an Arduino provides?

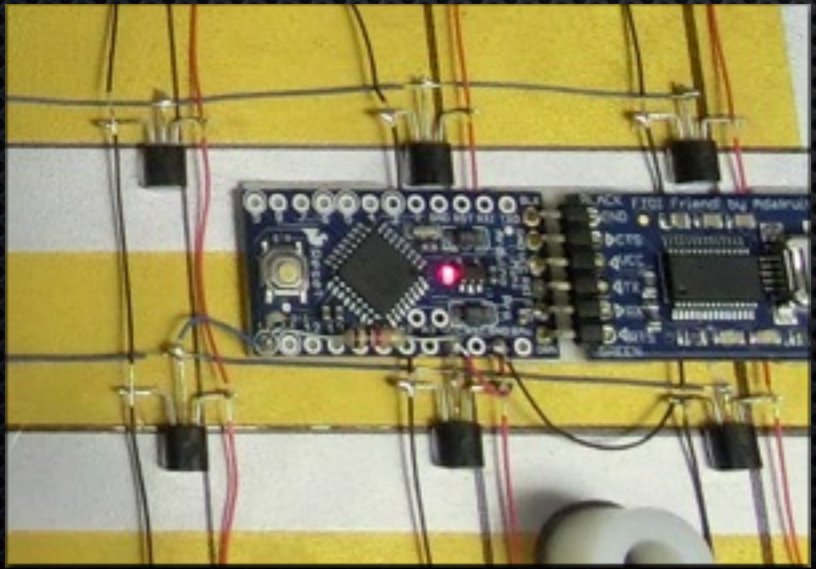

DURACELL


Arduino
www.arduino.cc
sparkfun



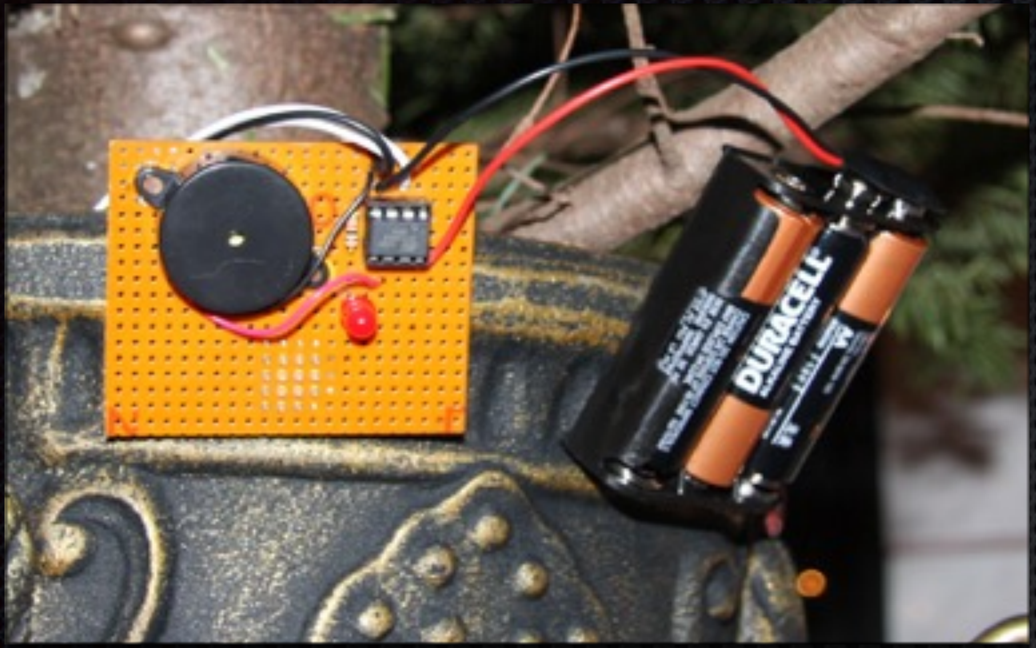

+5
GND
S A
S B
S sync
O sync
GND
V+

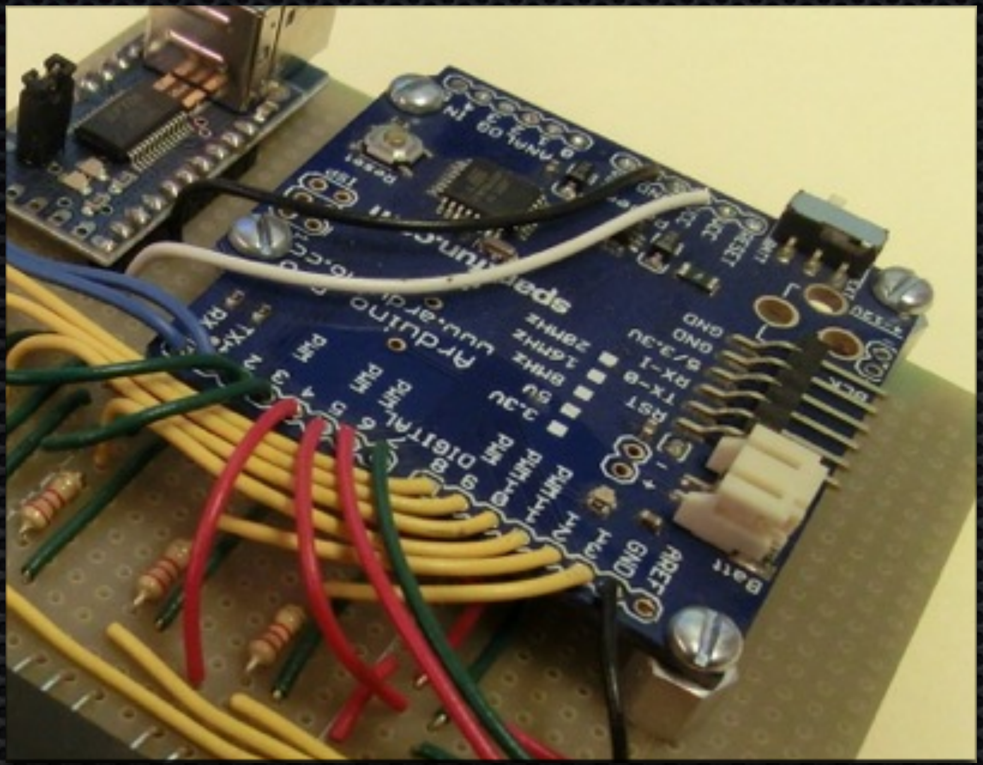
R3
CONTROLLER
C1 0.1uF
H-BRIDGE

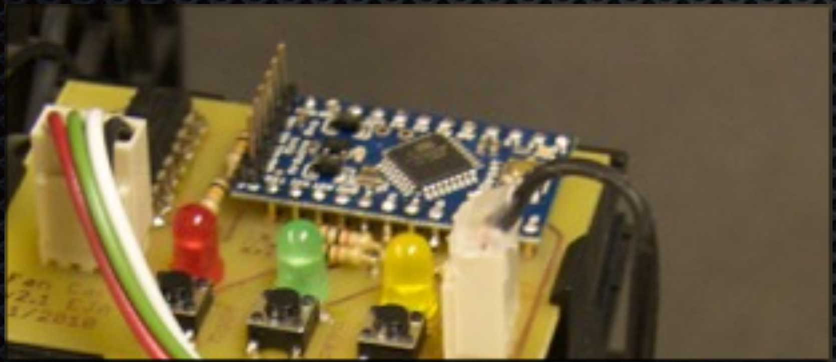


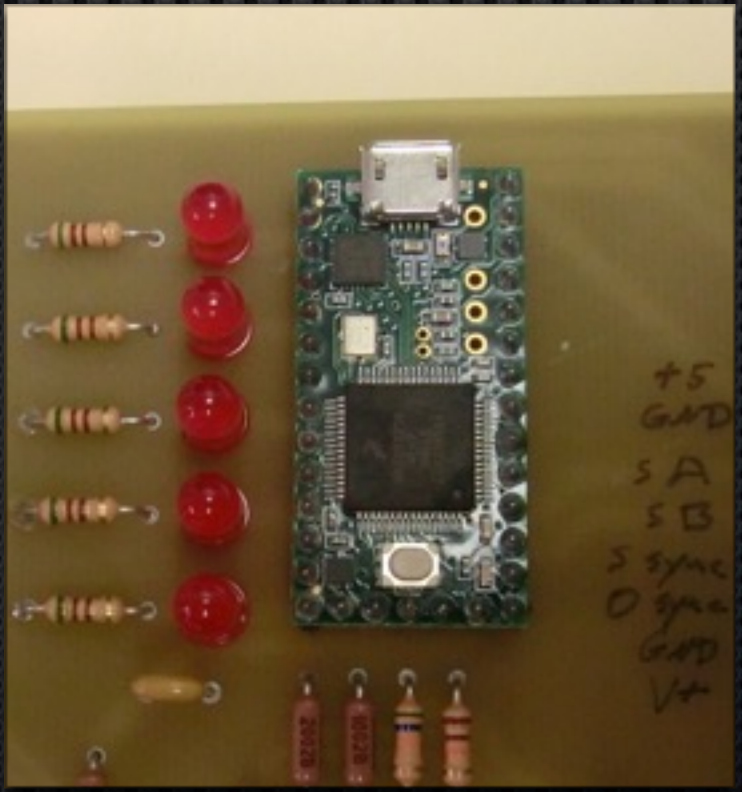
“Arduino” goes way beyond just Arduino.

- There are a lot of clones and other footprints that work within the Arduino IDE.
- Most Atmel microcontrollers are programmed by SPI. Arduino can communicate via SPI.
- You can program an Arduino to program other microcontrollers!
- You can still use the standard Arduino IDE, so it’s still “Arduino-easy”.

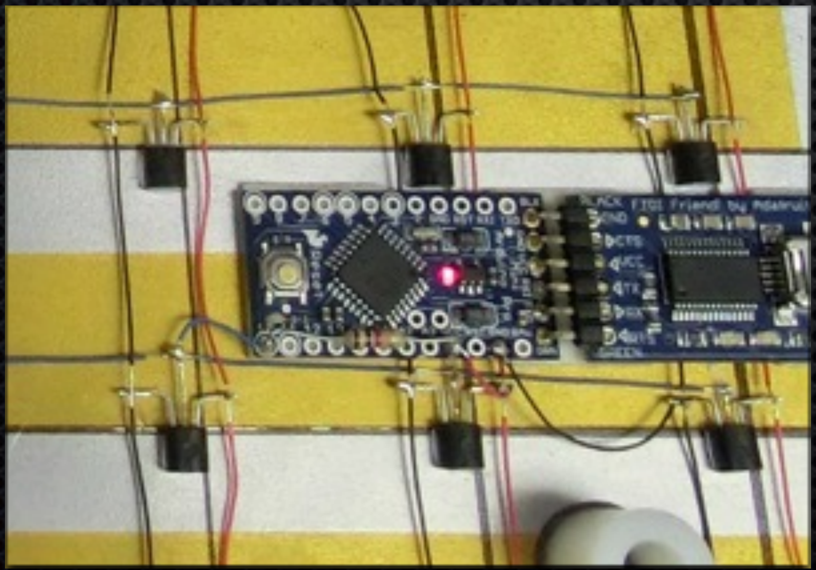

DURACELL


Arduino
www.arduino.cc
DIGITAL
PWM
GND
AREF
Batt
RST
3.3V
5V
8MHz
16MHz
20MHz
TX-0
RX-1
5/3.3V
GND
Reset



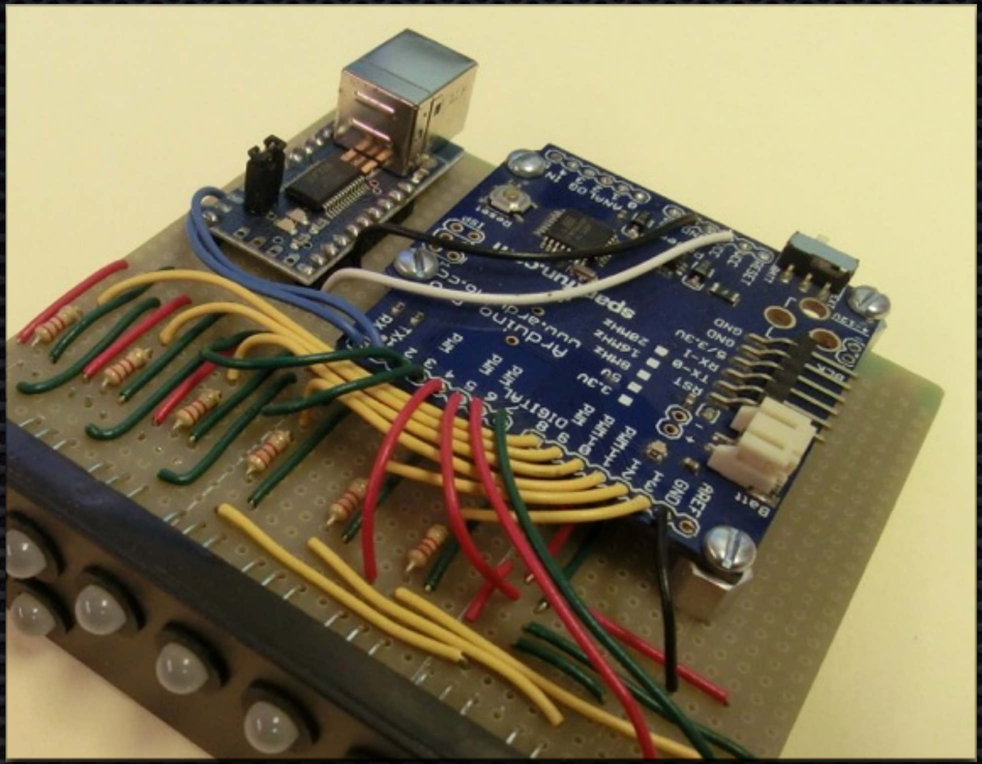

+5
GND
S A
S B
S sync
O sync
GND
V+

R3
CONTROLLER
ATTINY84
20PU
C1 0.1uF
H-BRIDGE


FTDI Friend by Adafruit

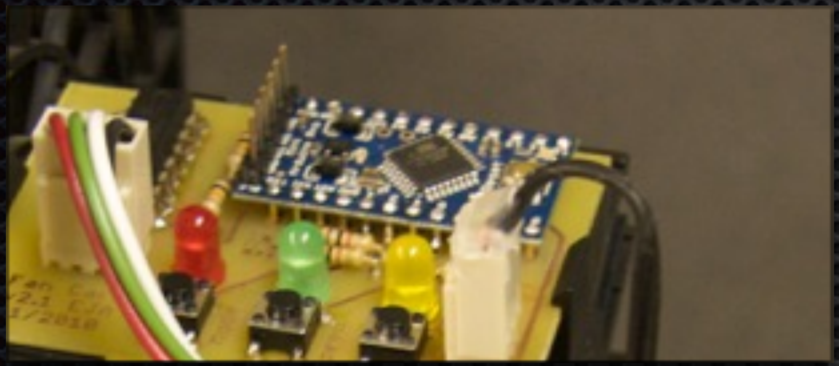
Clone!

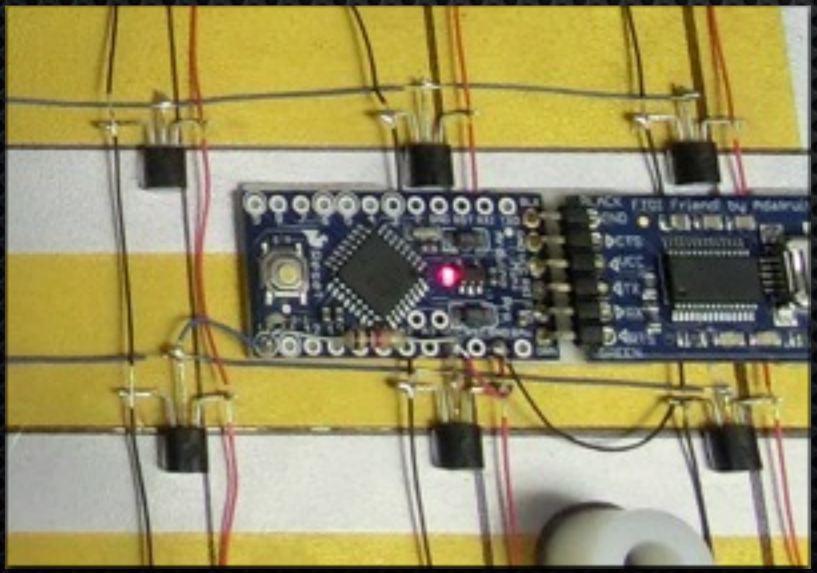
- Cheap Arduino Clone (sparkfun.com)
- Same footprint as Arduino, half the cost.
- It’d be nice to have this in a breadboard-friendly package…

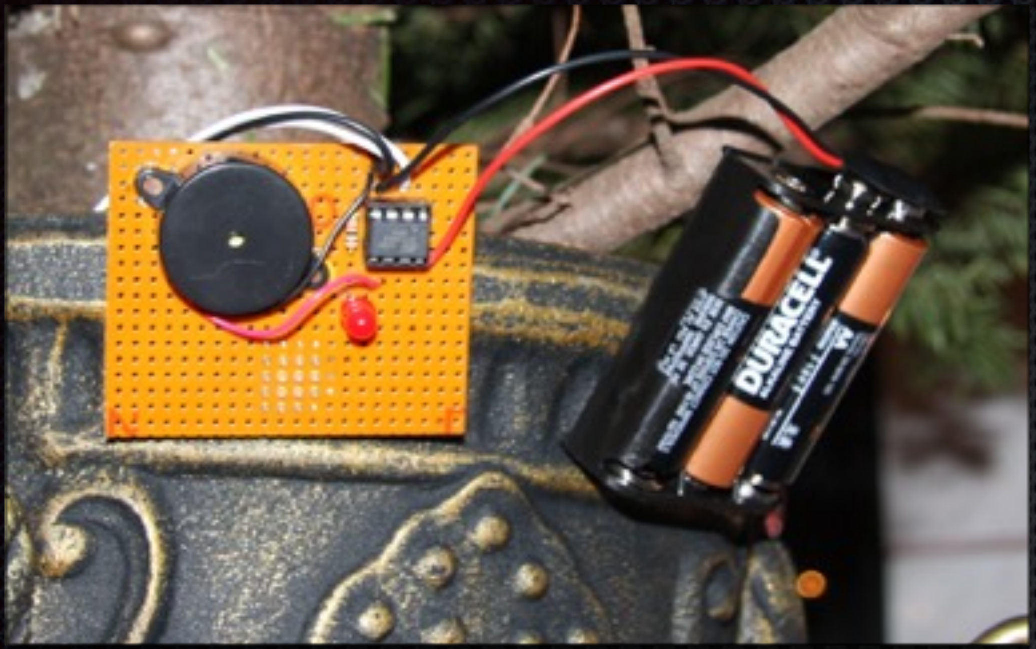


Arduino Pro Mini

- same chip (ATmega328)
- breadboard-friendly layout
- cheaper than stock Arduino
- Slightly fewer pins
- USB—serial adaptor missing
- obsolete now, the “Pro Micro” is the same form factor with USB.







ATtiny45 or ATtiny85

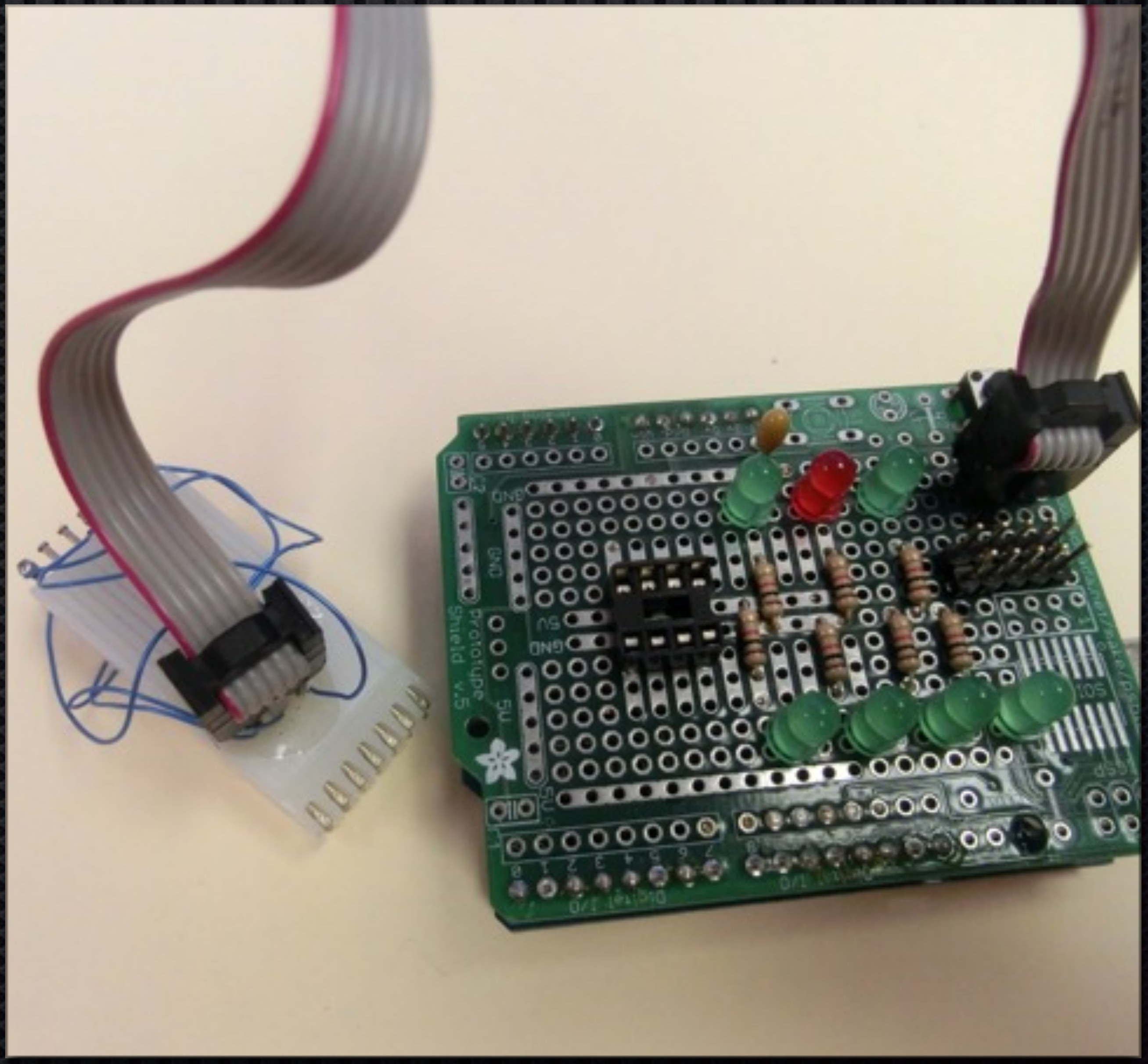
- 5 i/o lines
- 4k or 8k memory
- lower power requirements
- $1.35 each (cheaper in bulk)

ATtiny84

- Same internal capabilities as the ATtiny85, more i/o pins.

My programmer

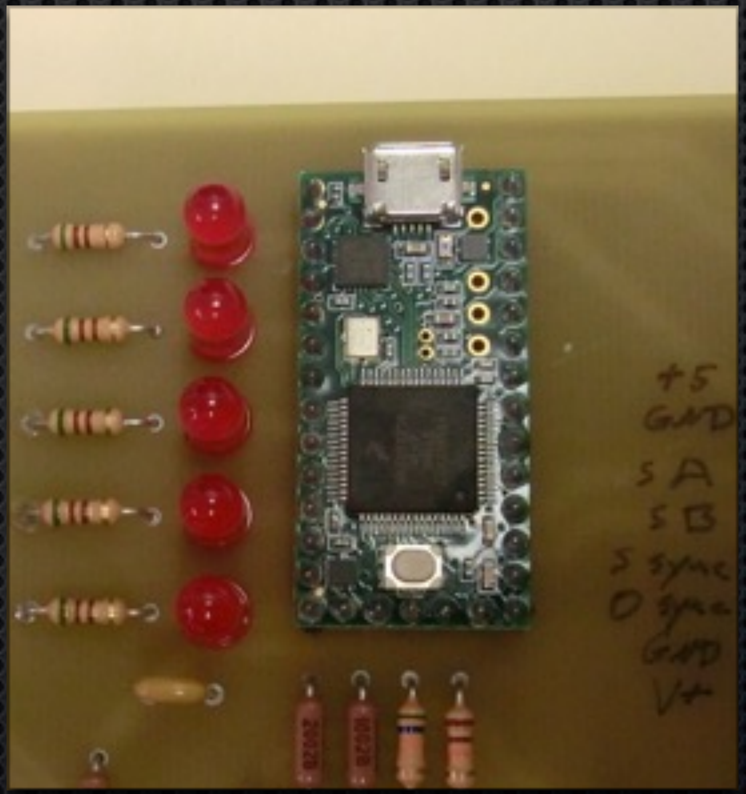
- There's a "real" Arduino under there…
- 8-pin chips (ATtiny85) can be plugged directly in and programmed
- 14-pin (ATtiny84) I can grab with the chip-clip and program in-circuit.



Teensy 3.1/3.2

- ARM Cortex-M4 processor
- 6x speed, 8x memory
- 16-bit A/D inputs
- 12-bit D/A output
- dedicated hardware interfacing lines

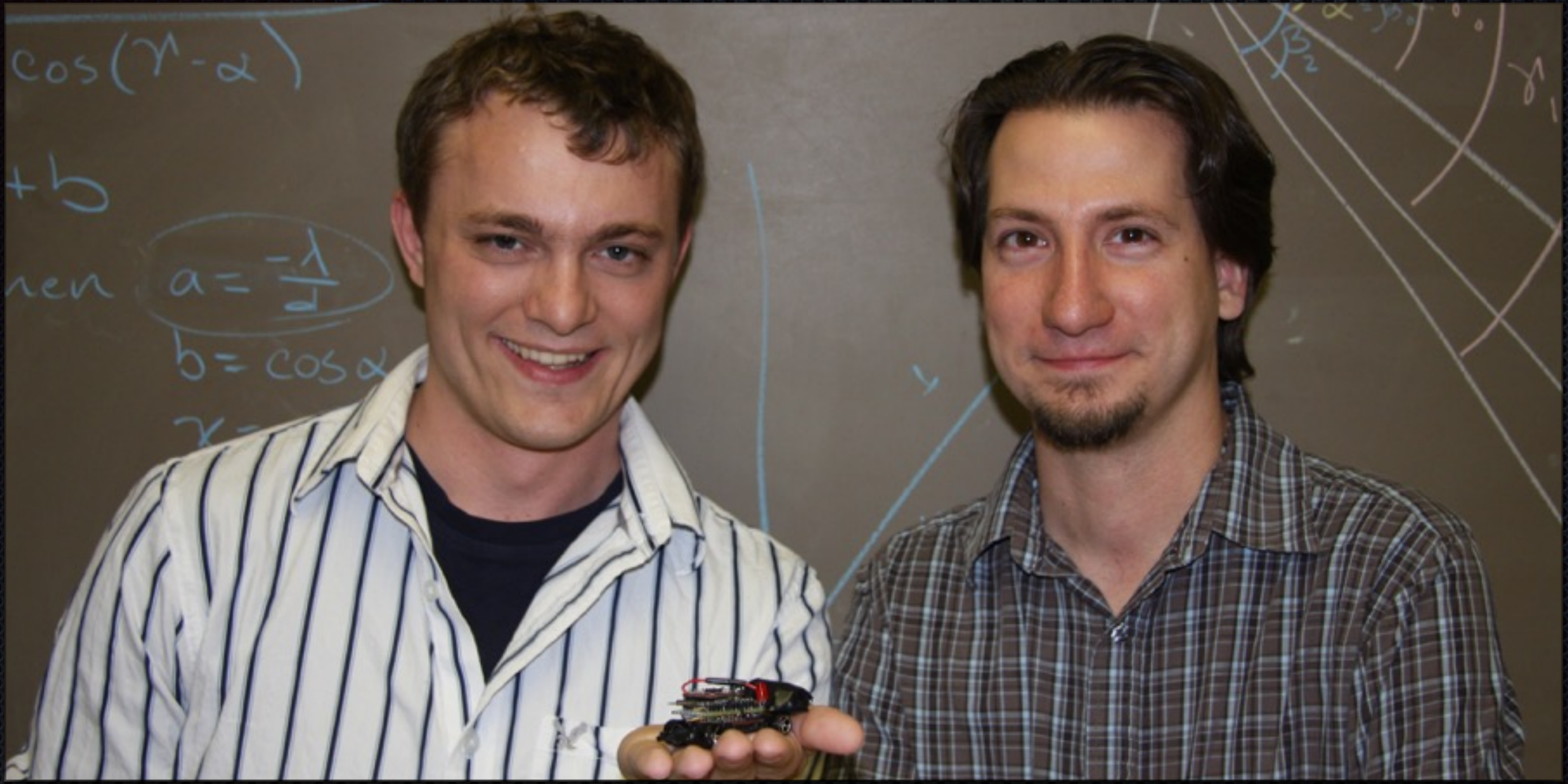
- Built-in real-time clock
- Built-in USB
- $20



Summary

- Microcontrollers are best when used as single-purpose dedicated hardware.
- Don’t get carried away.
- There are a lot of non-Arduino options: pick what’s best for your application.

Finally…



Best practice for teaching:

Face it: many students are smarter and more creative than us. Supply cool tools and stand back.