

OpenLCB

- “Layout Control Bus” huh?

- “OpenLCB” huh?

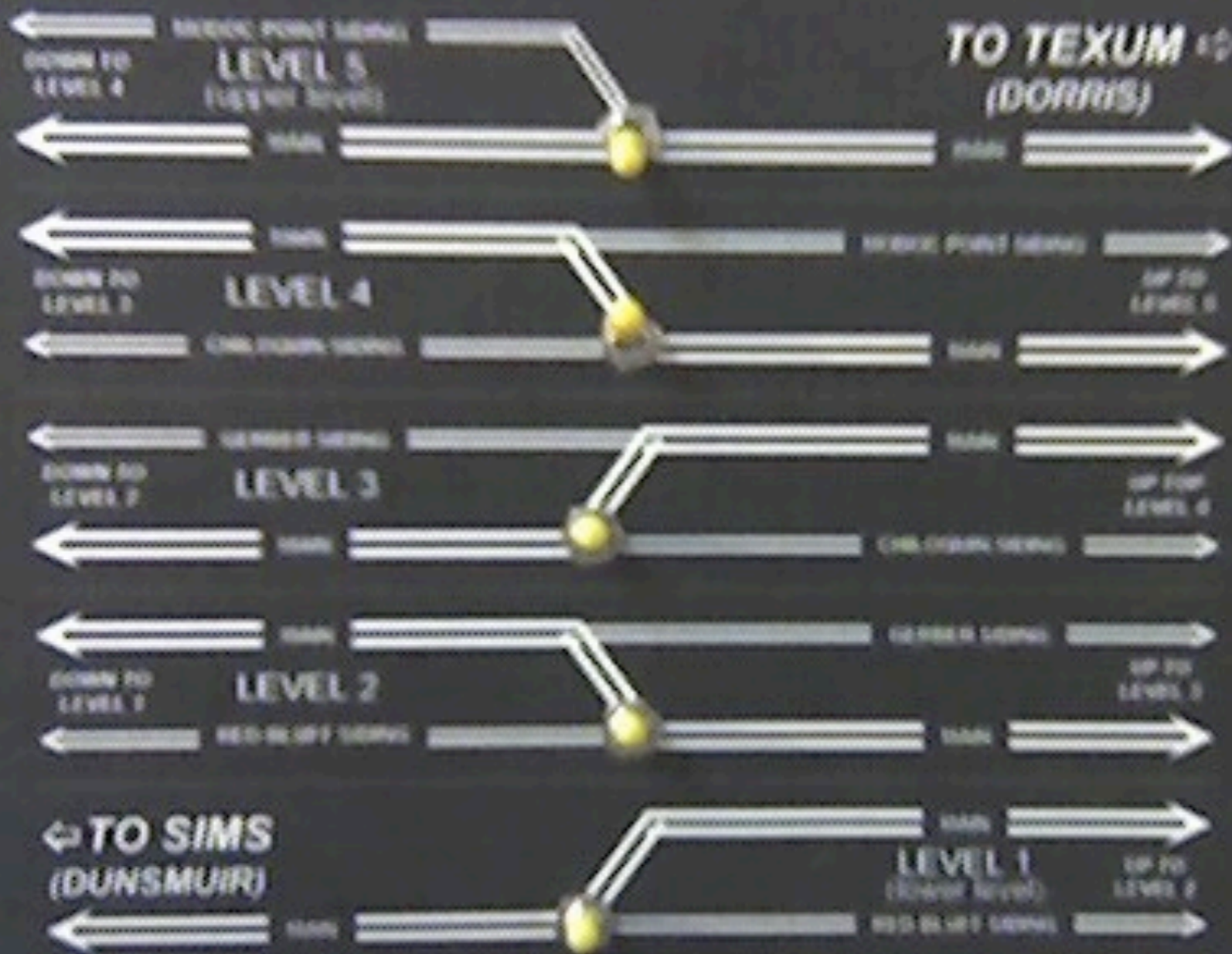
Make simple things simple, powerful things possible.

Technology for the next 30 years

- What have we got so far?
- Where are we going?

— HELIX TURNOUT CONTROLS —

Down Position For Straight – Up Position For Diverging (Crossover)



Pretty simple switches. Just wire to Tortoises. Except the owner wants two panels. And to not allow throw under train....



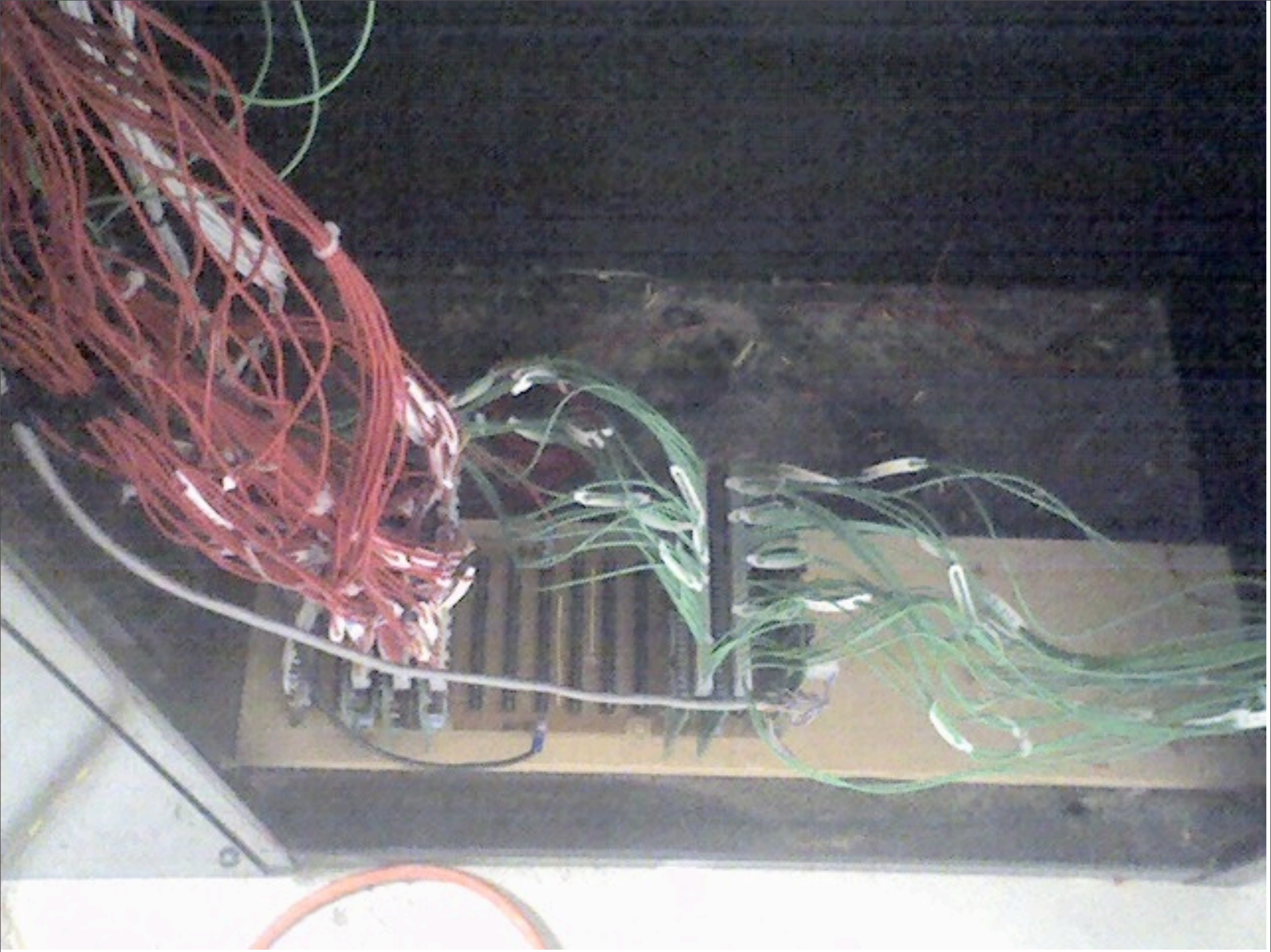
Tuck your shirt in before taking pictures of shiny things. And one button has to do various things, which overlap with other things....

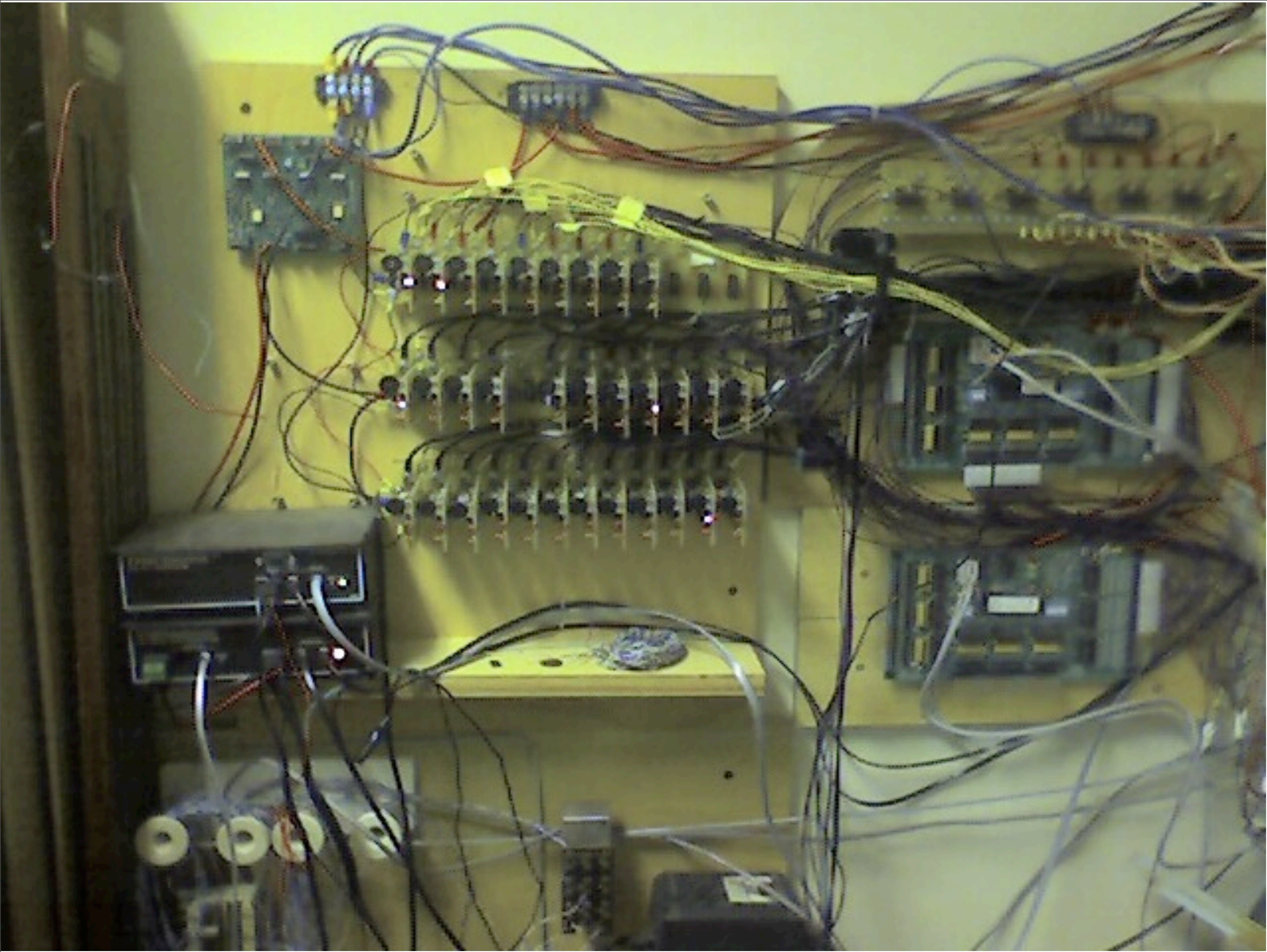


The turnout drivers at the other end

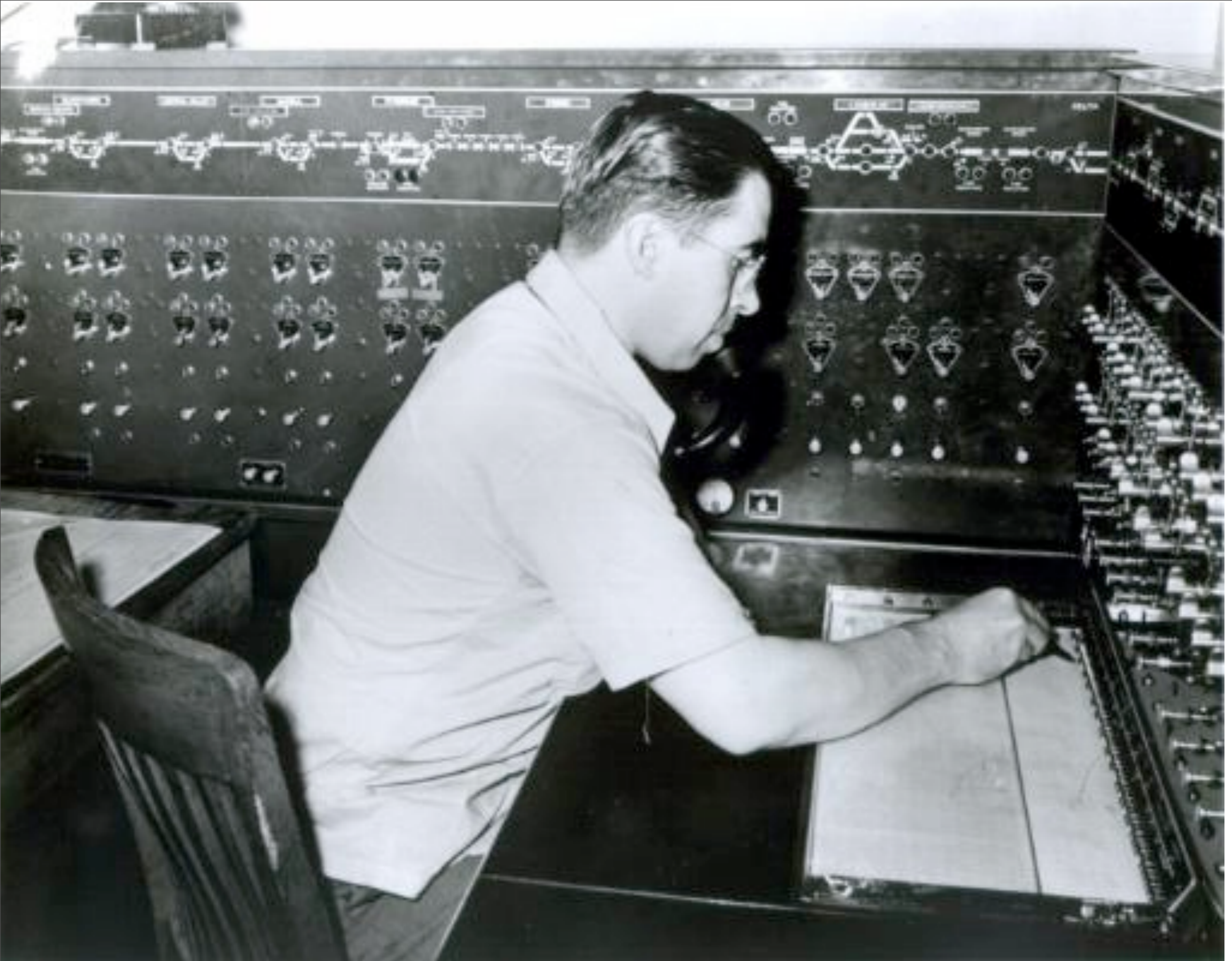


But you need wires running to where the magic happens





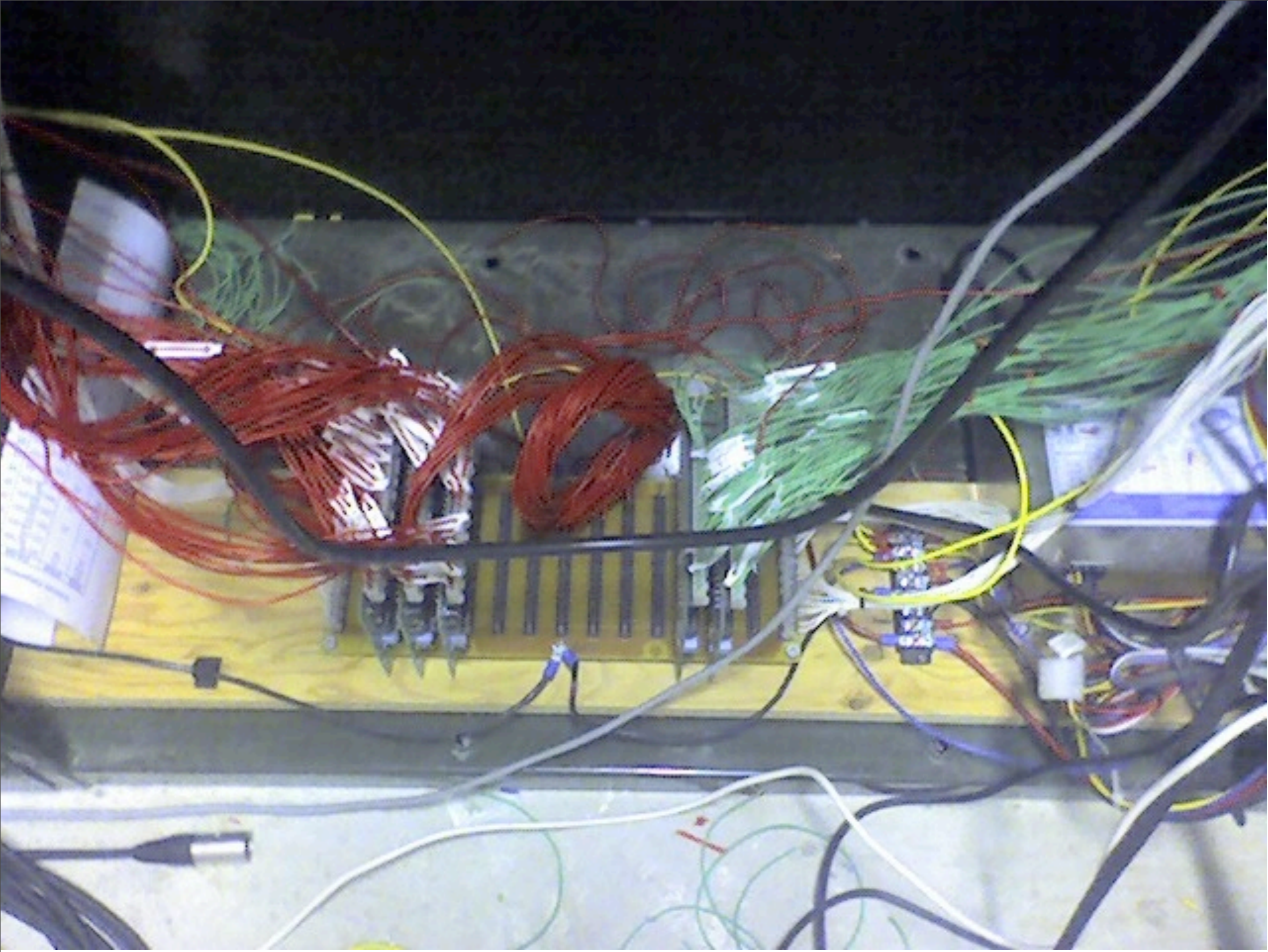
Point out the nodes. Command stations are separate. How the buses (both DCC, C/MRI, and NCE)

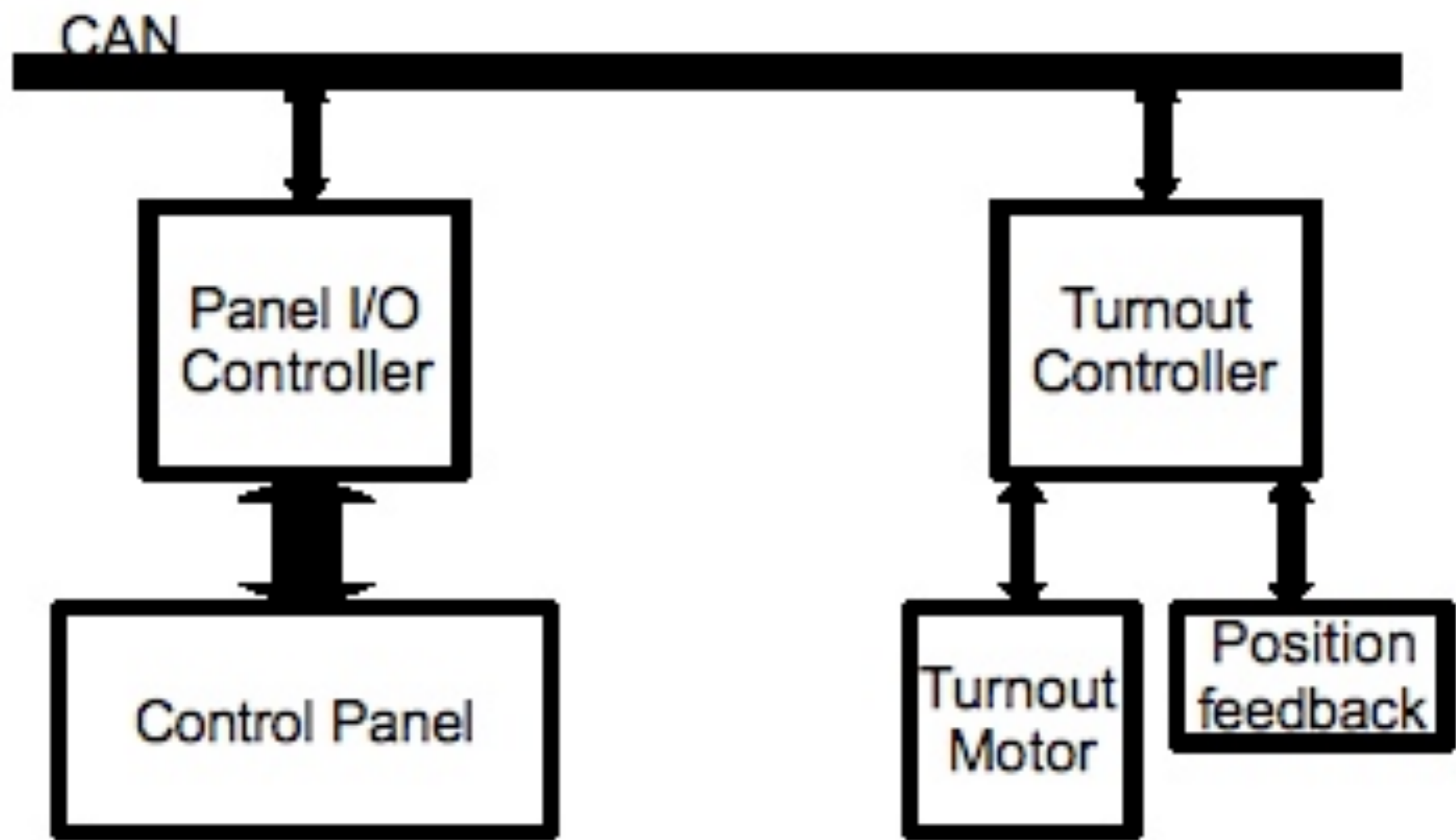


Some people model the railroad control system.

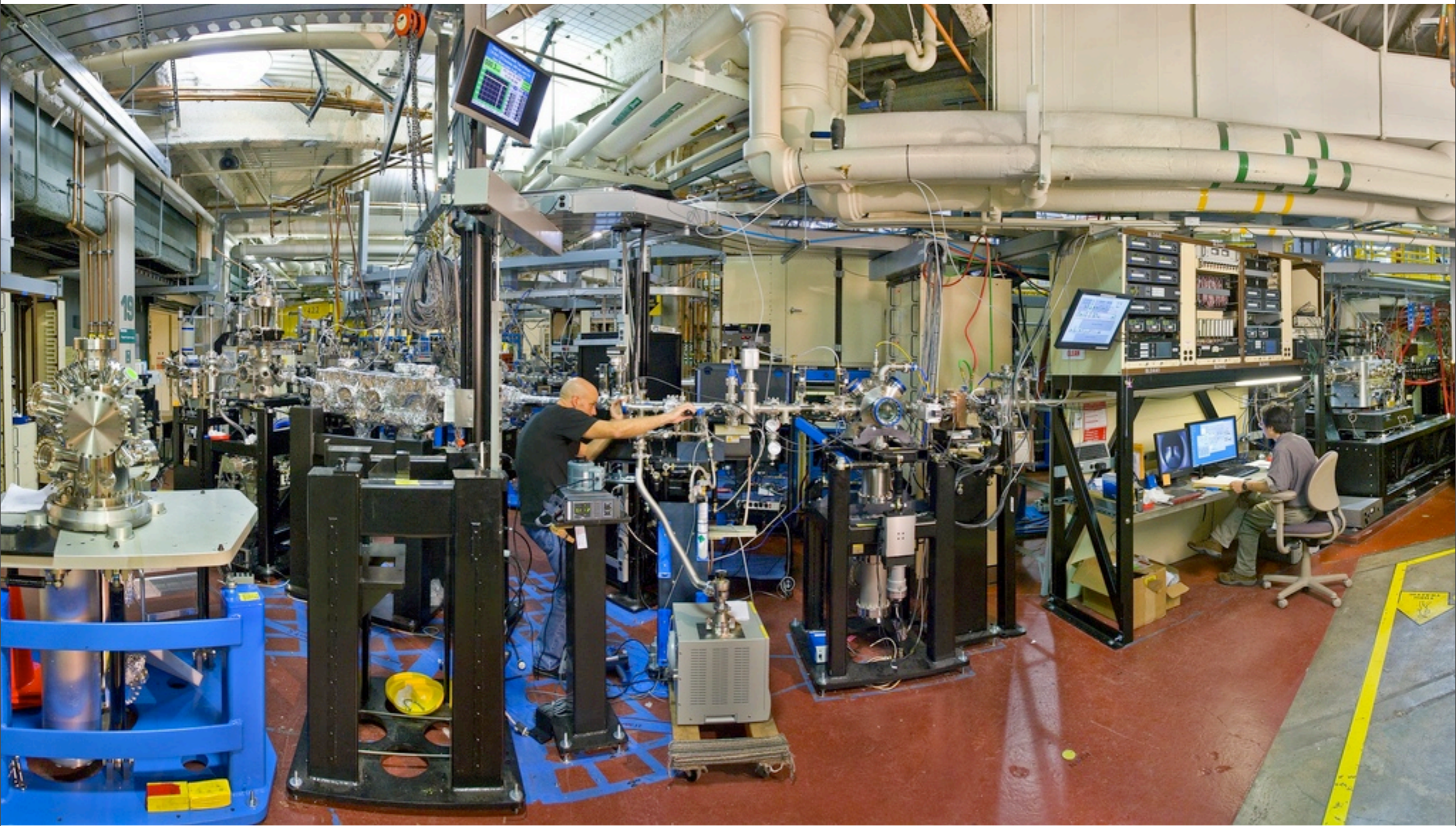


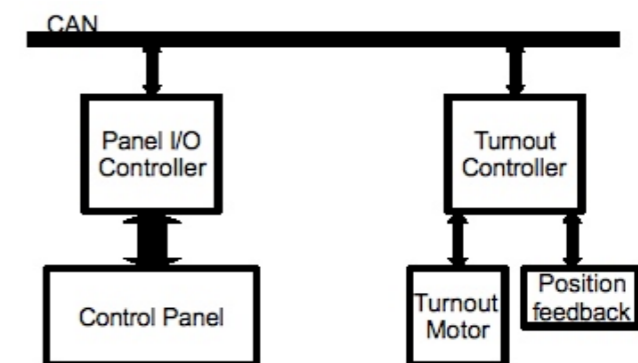
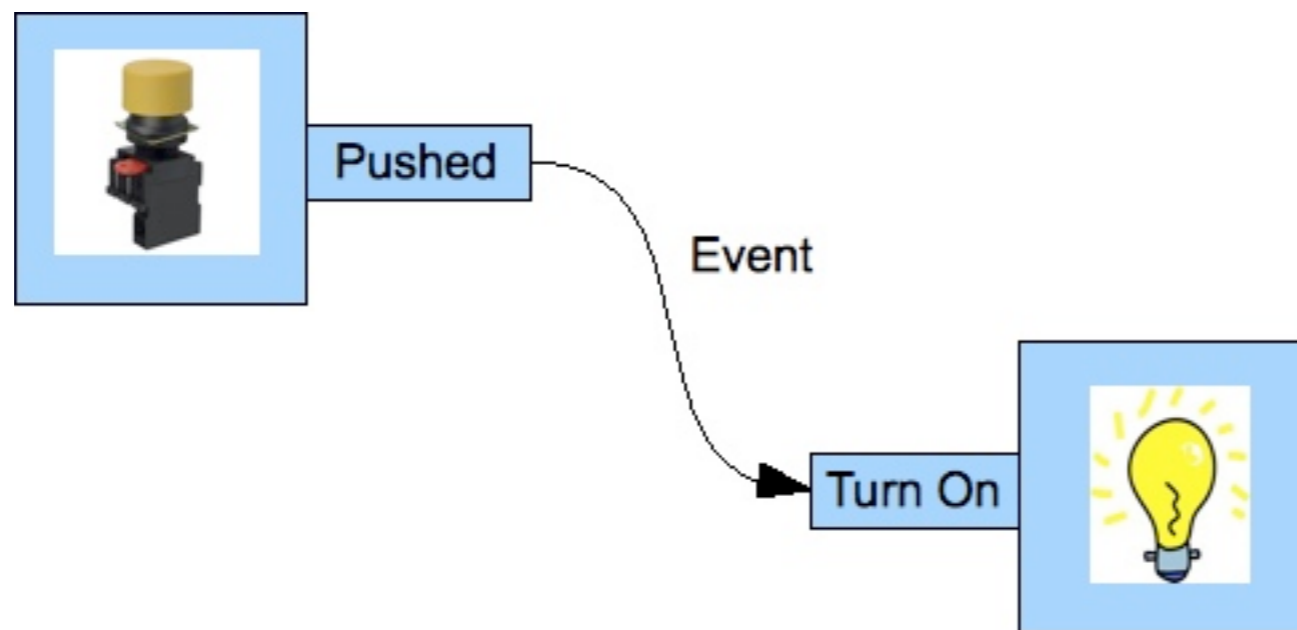
All those lousy lights....



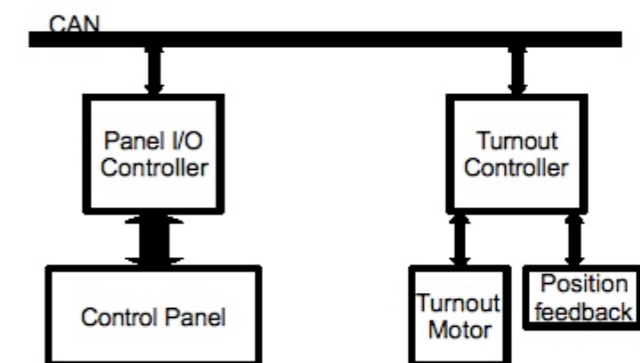
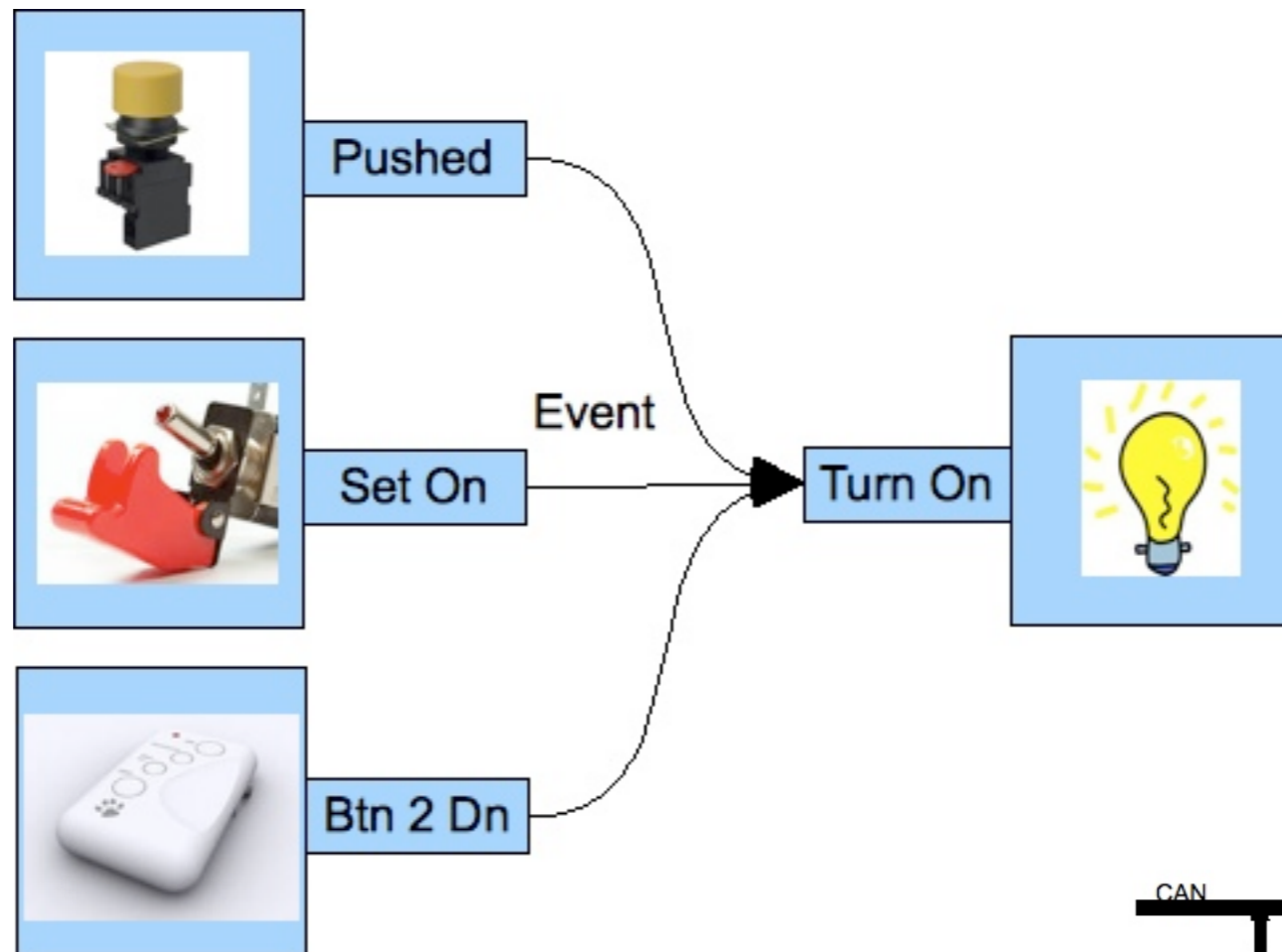


So here's where OpenLCB starts. These exist. Our interest is to push them to the next level. Why CAN?

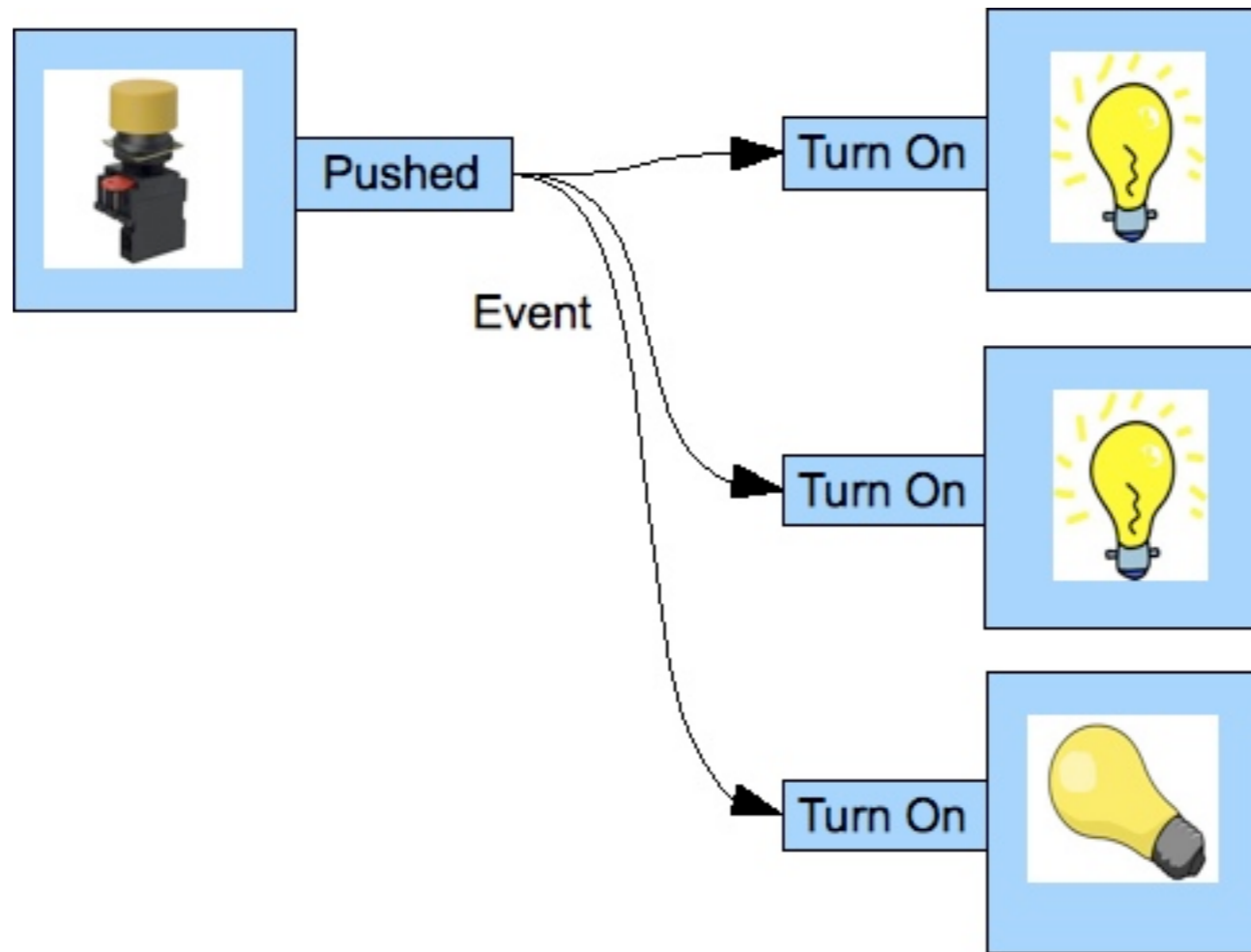




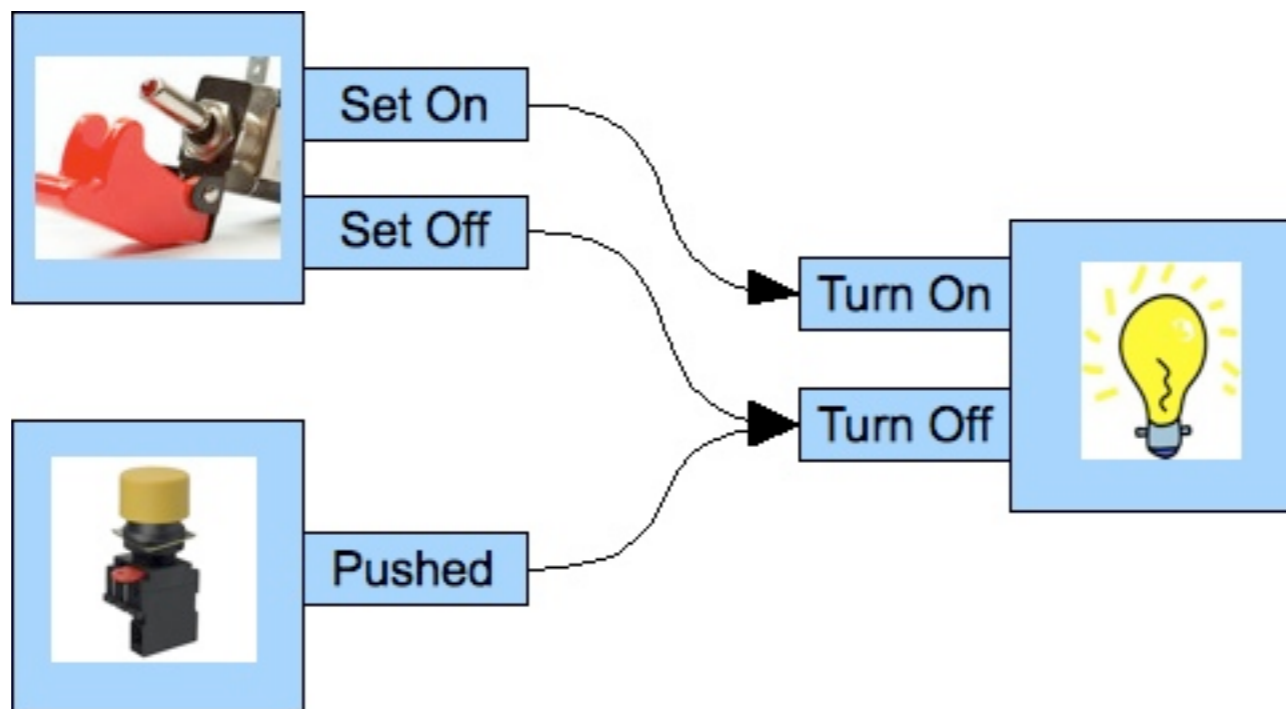
What that “bus” does is move information from one place to another. (Doesn’t matter where) Wires do that too, moving electrons, but these control systems send a message/number from one to another.

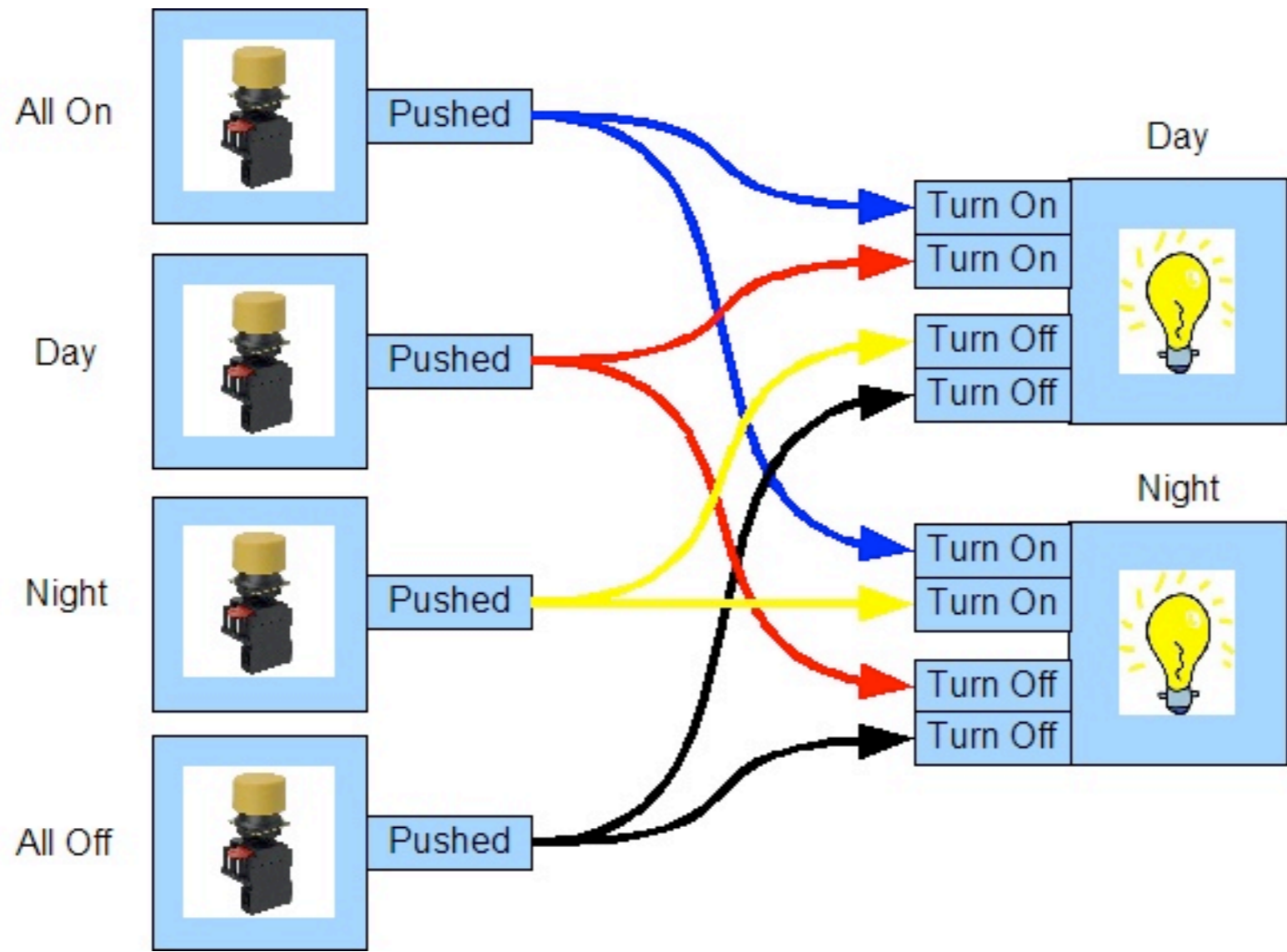


More than one can cause the same effect.



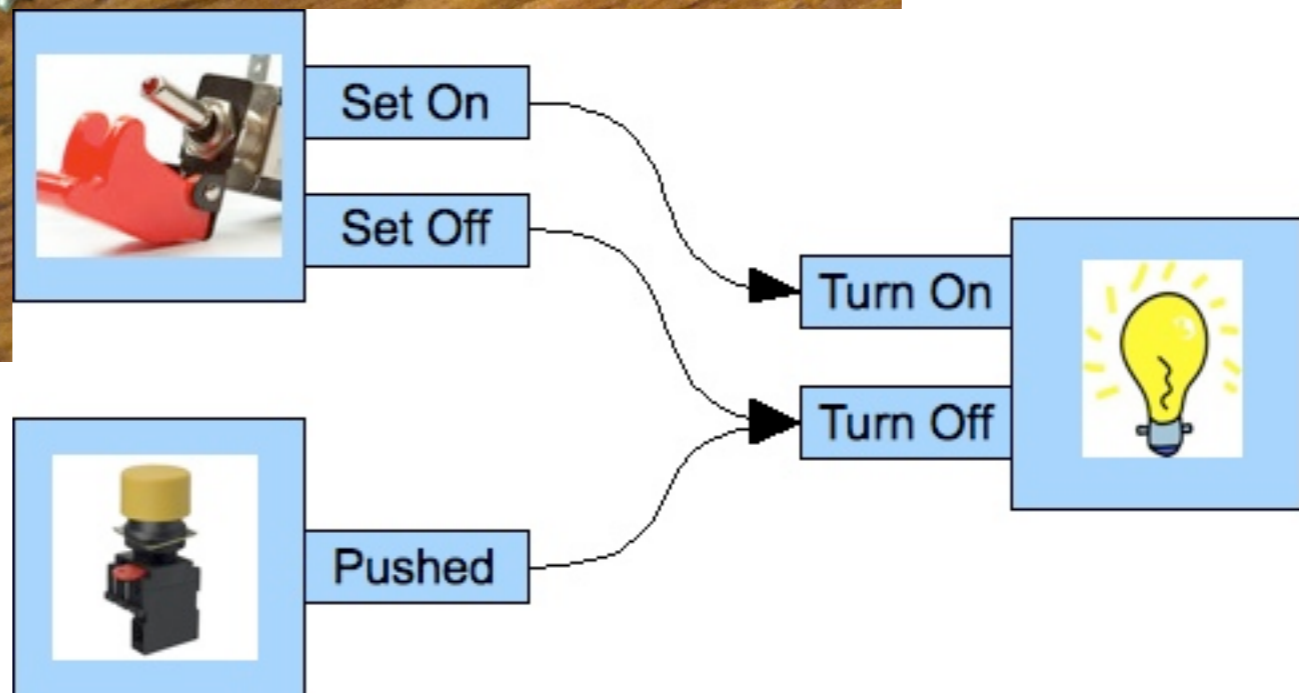
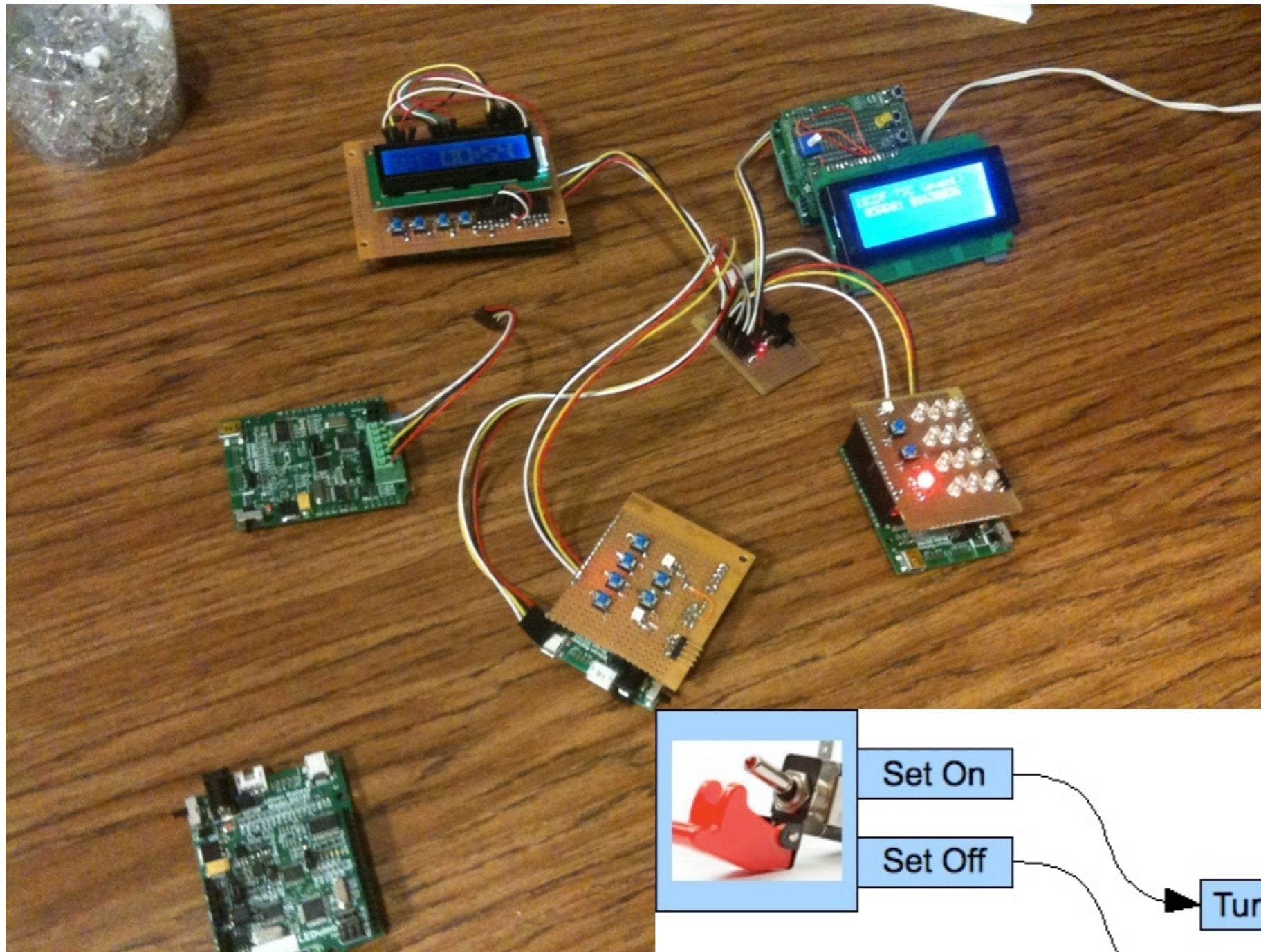
One event can cause multiple things to happen.



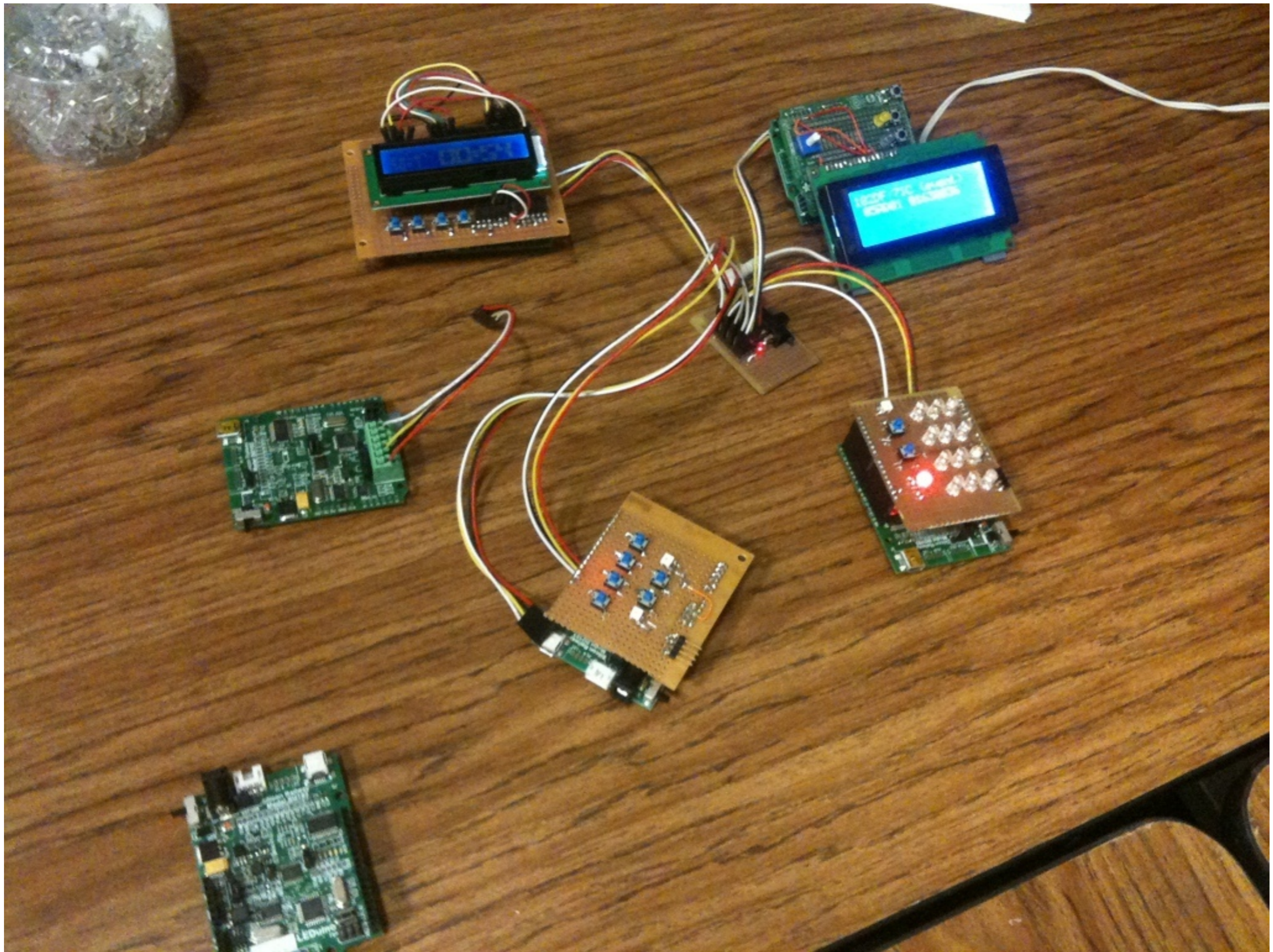


The colors have to be separate. Say why.

(Now do the demo)



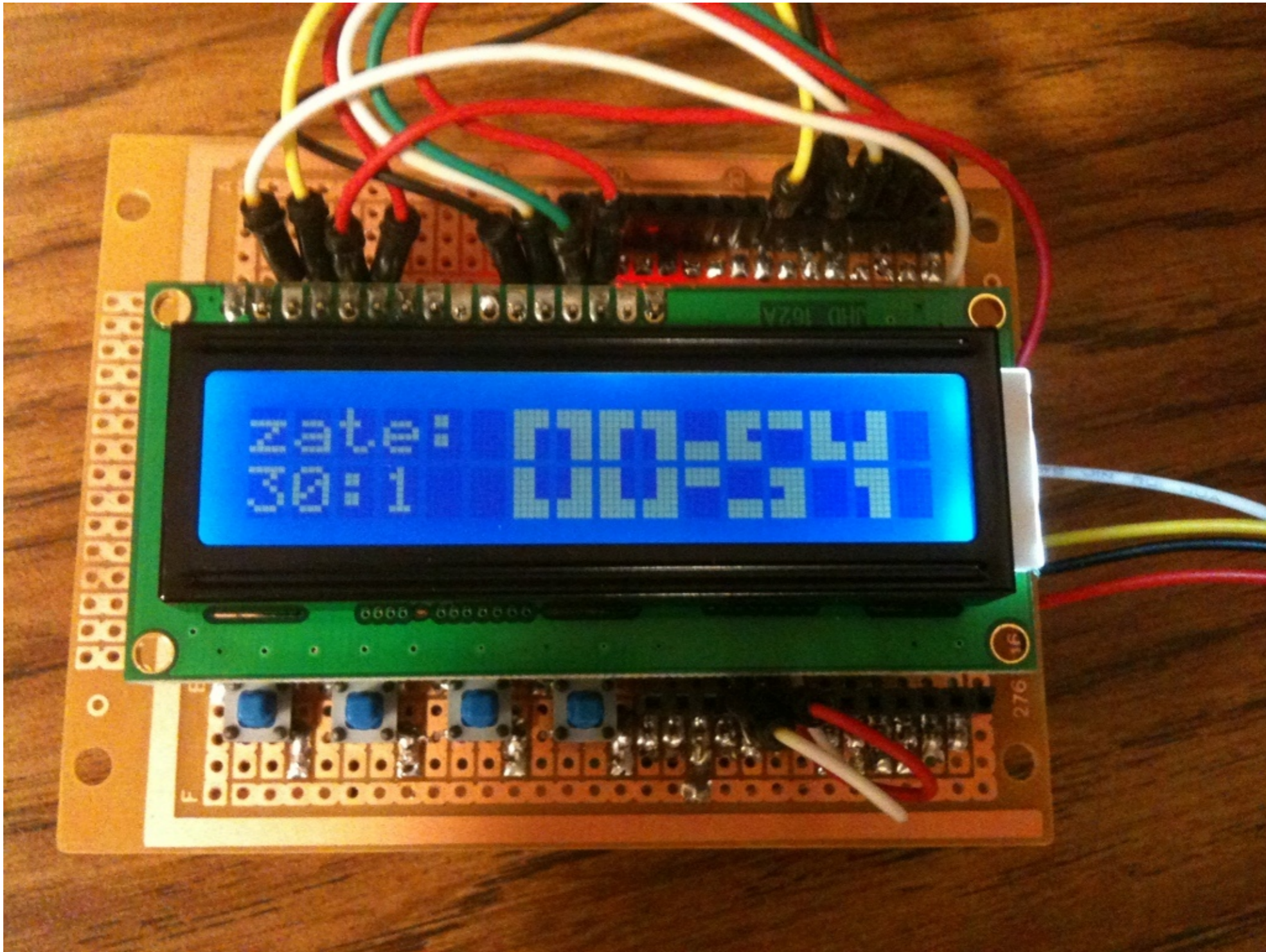
And do the Java demo too, don't forget



Hobbyist hardware – based on low-cost commercial processors

(now do the demo)

(now do the demo)

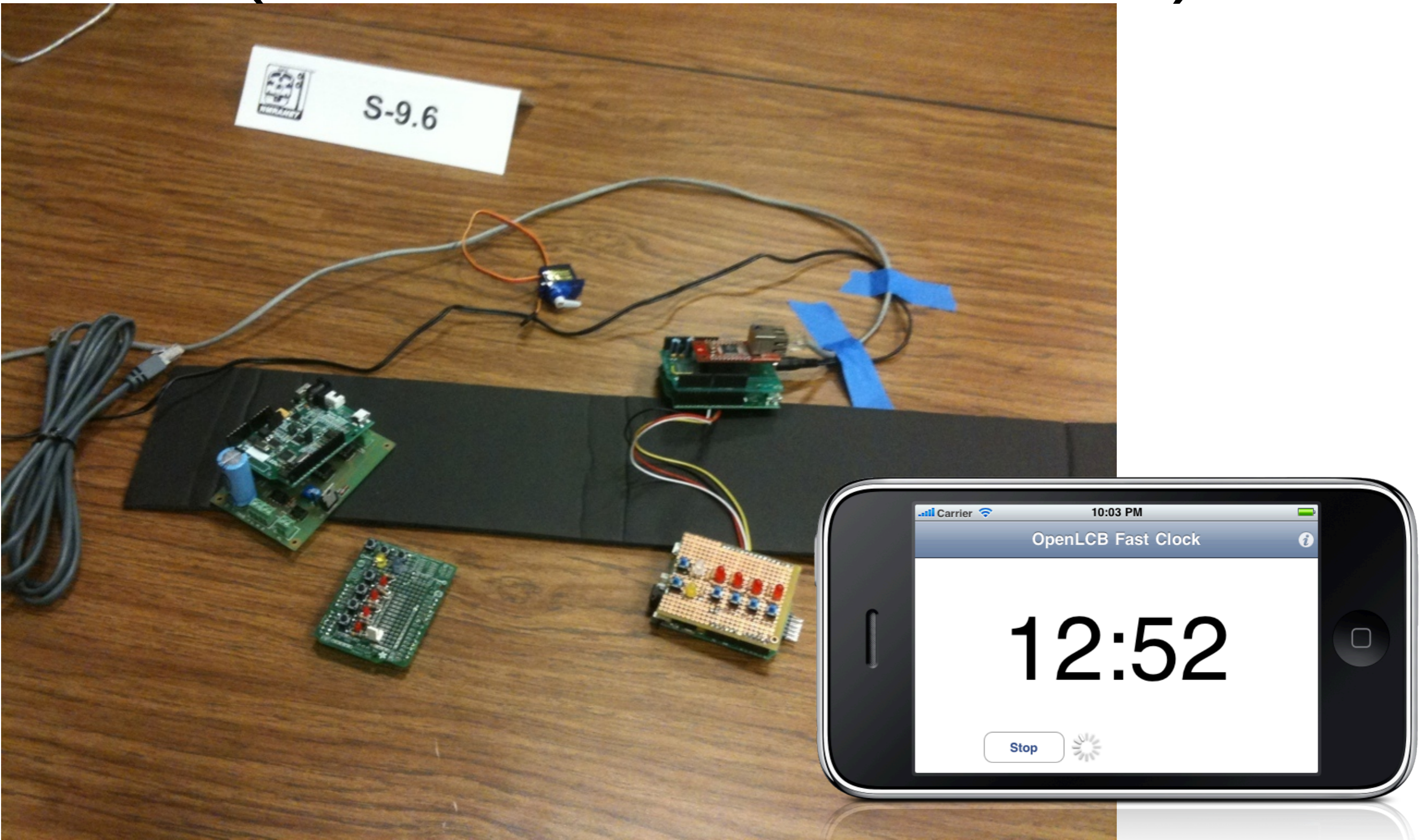


(Now do the demo)

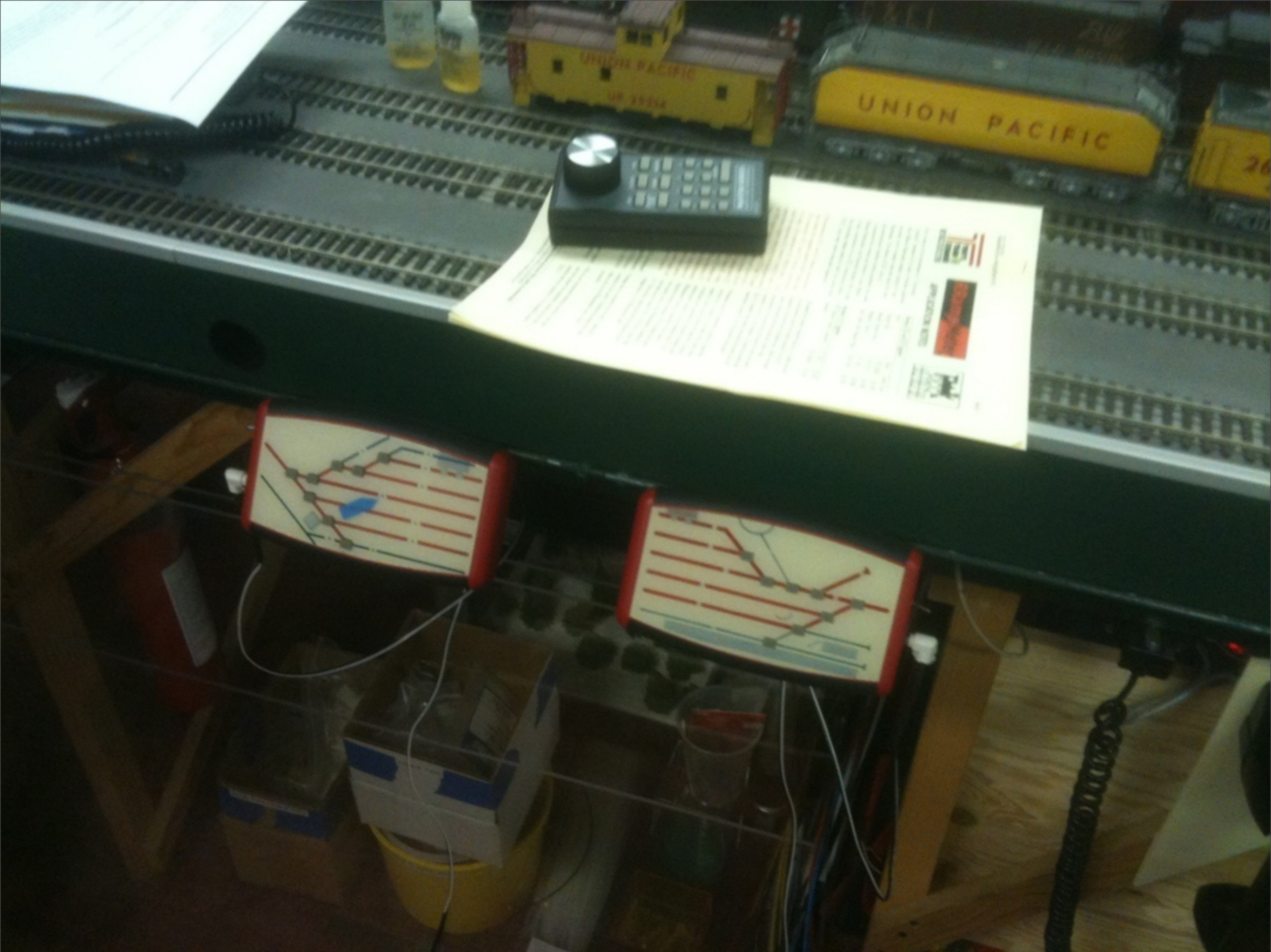


Try not to mention Ethernet. Something like “a few cables are better than lots of wire, but no cables at all is better”

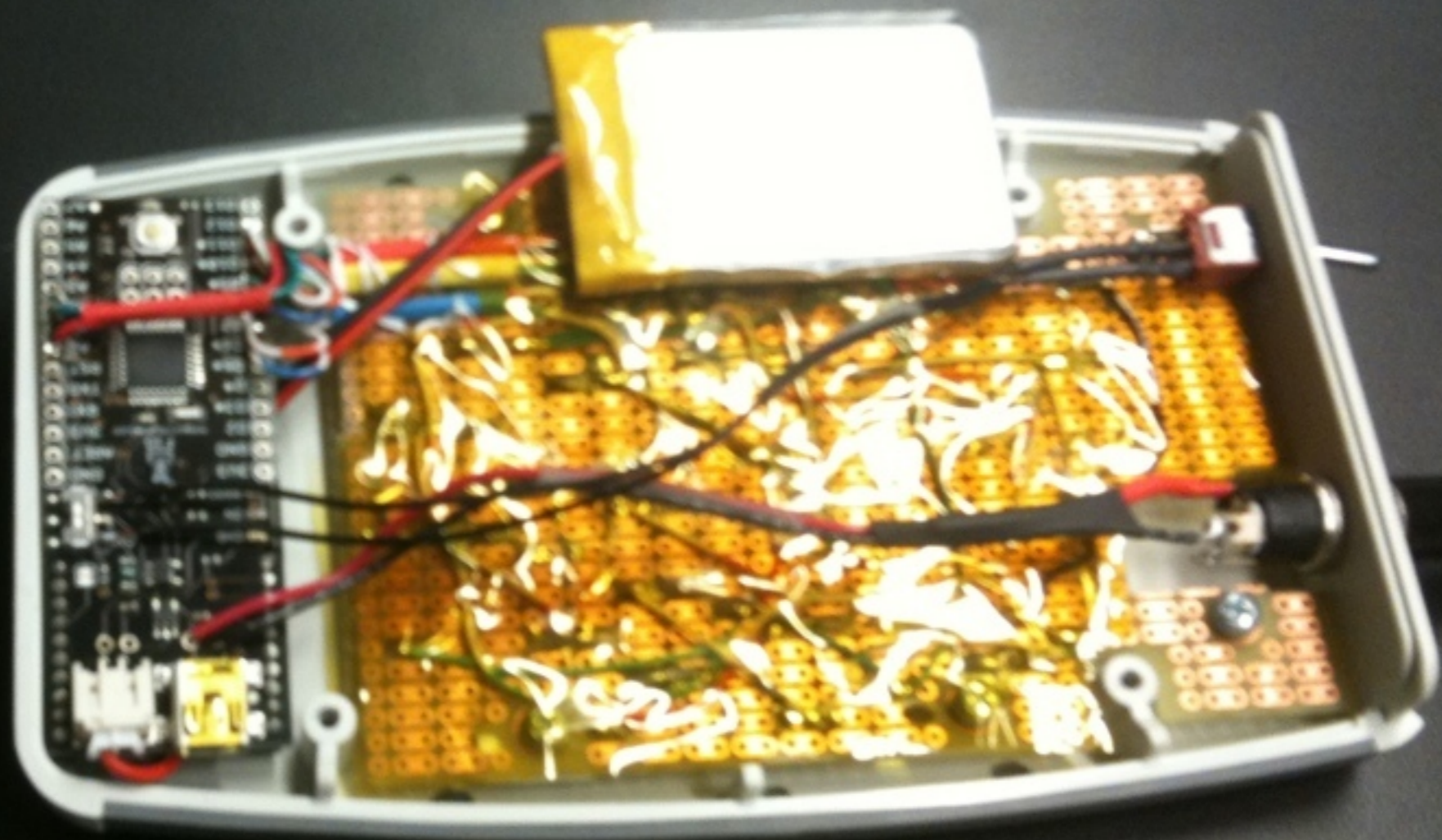
(Now do the demo)

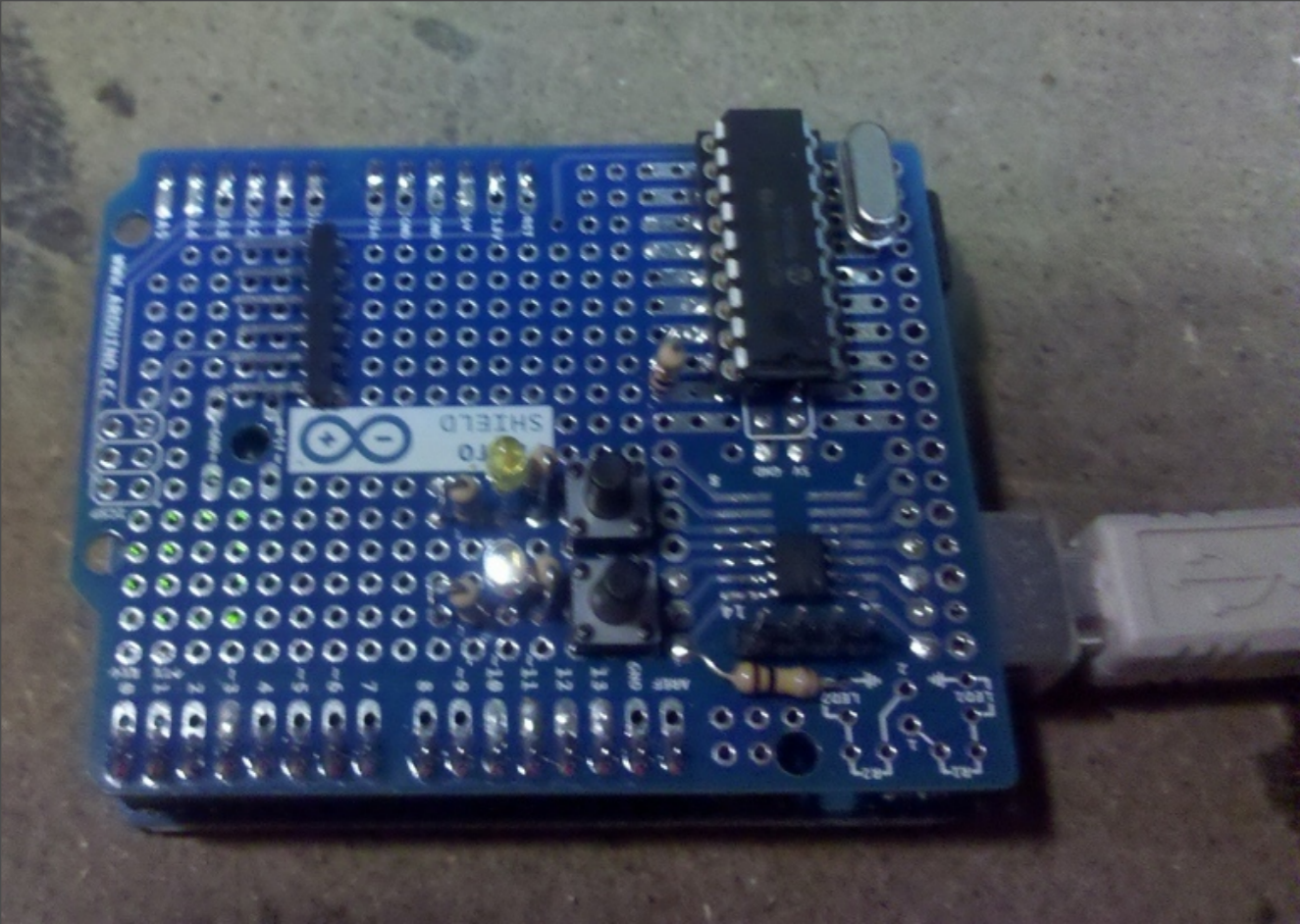


Try not to mention Ethernet. Something like “a few cables are better than lots of wire, but no cables at all is better”

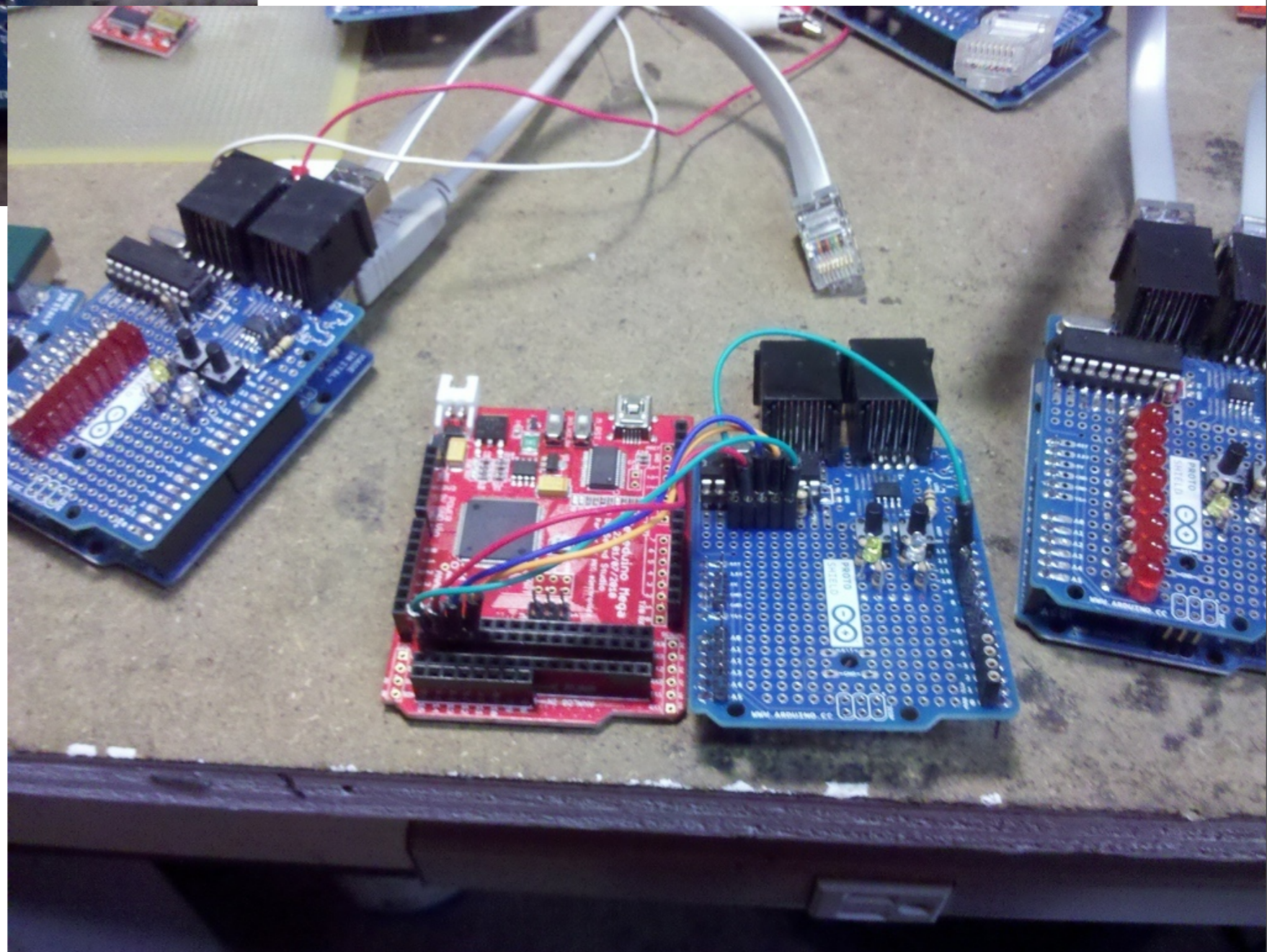
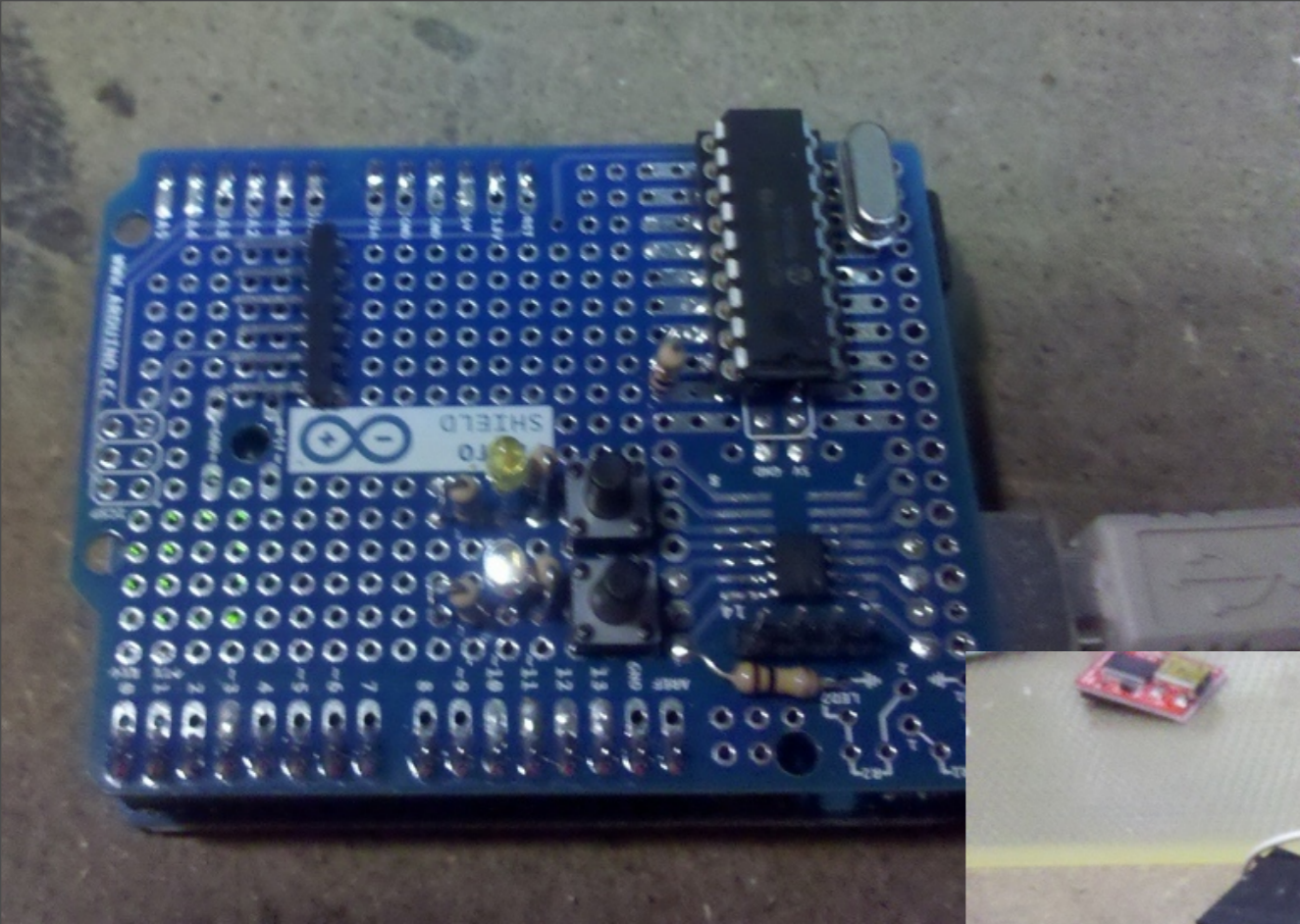




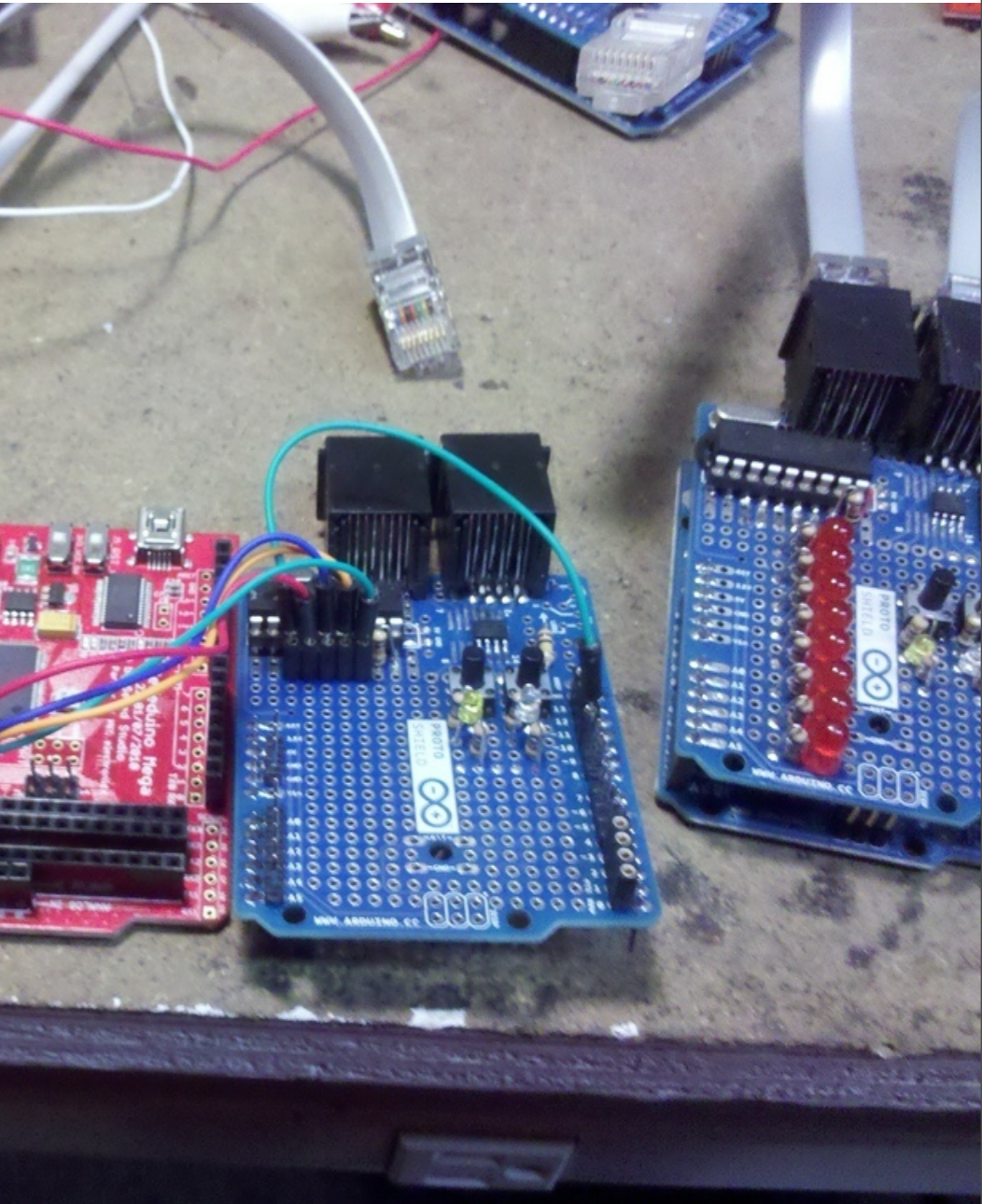
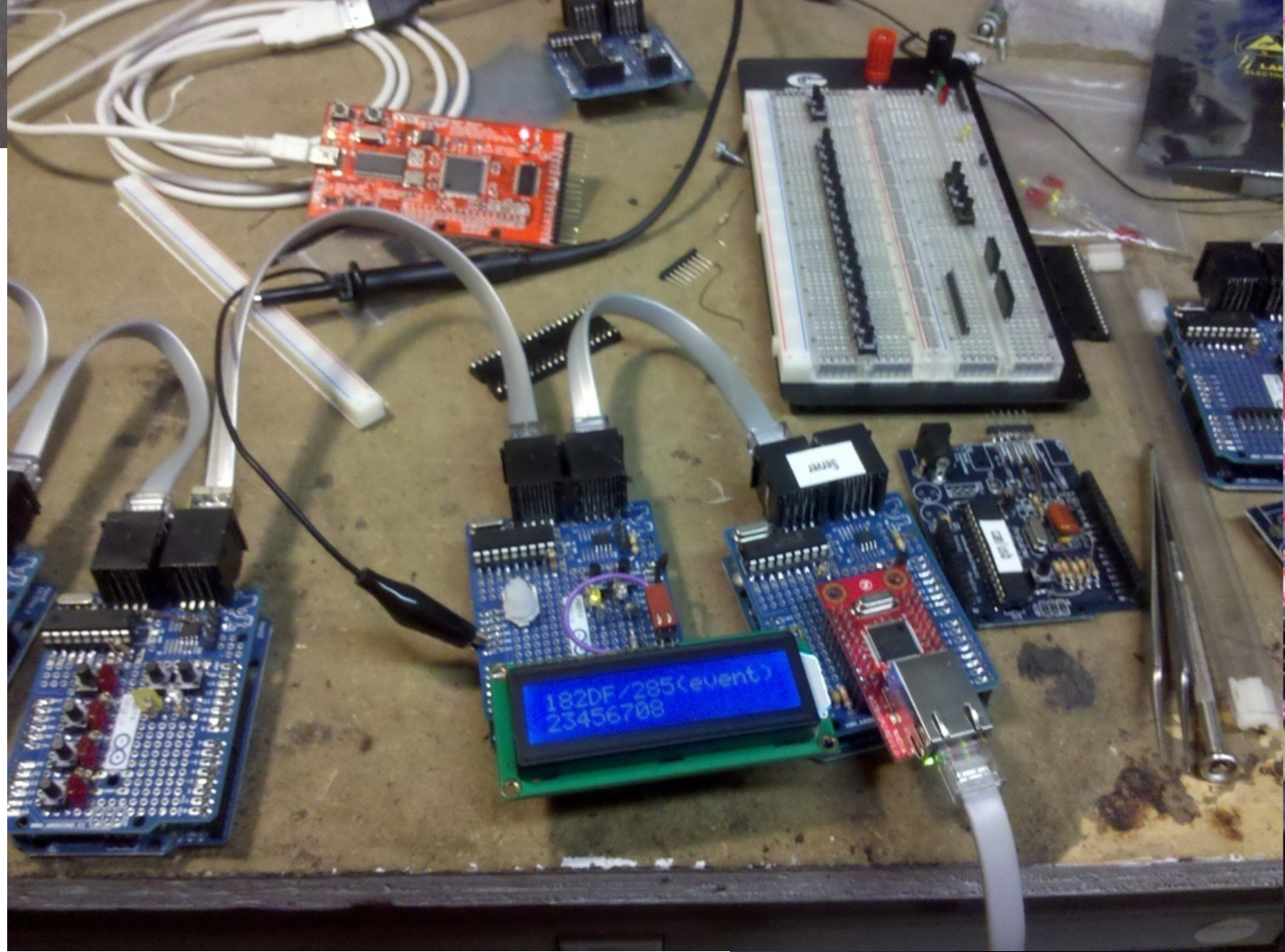
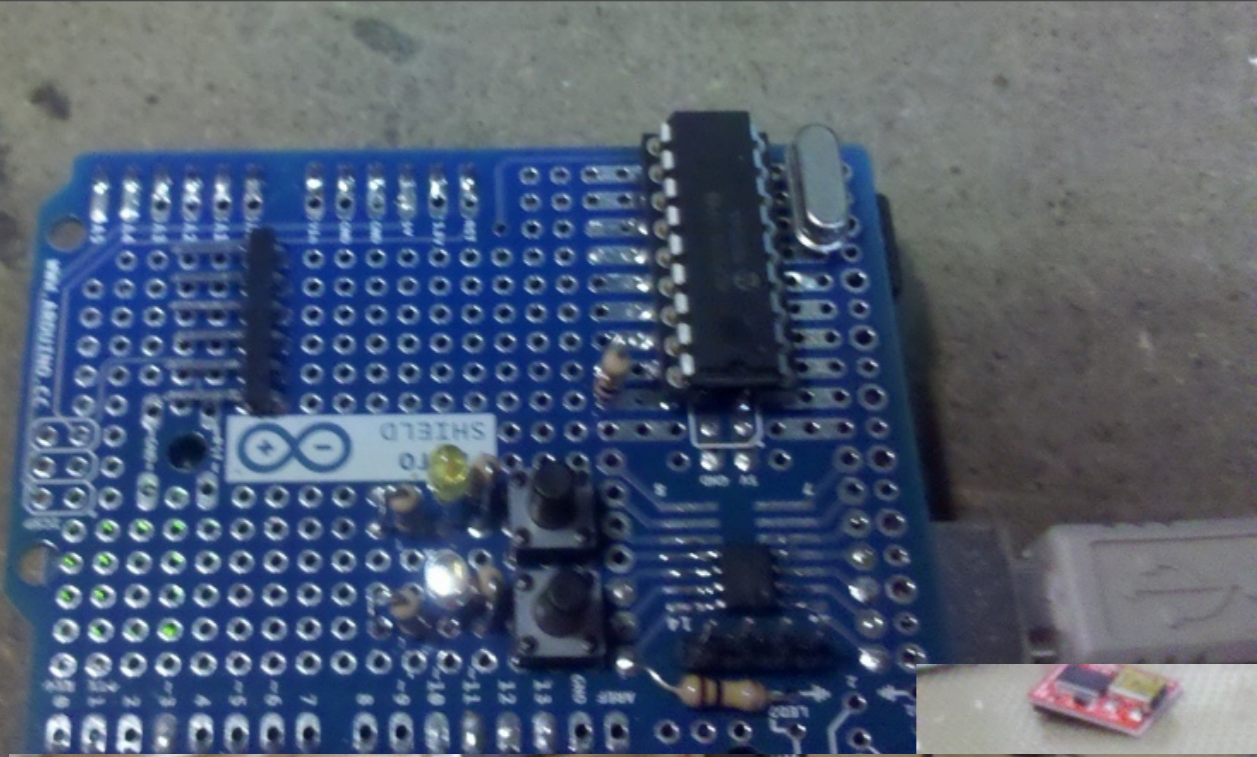




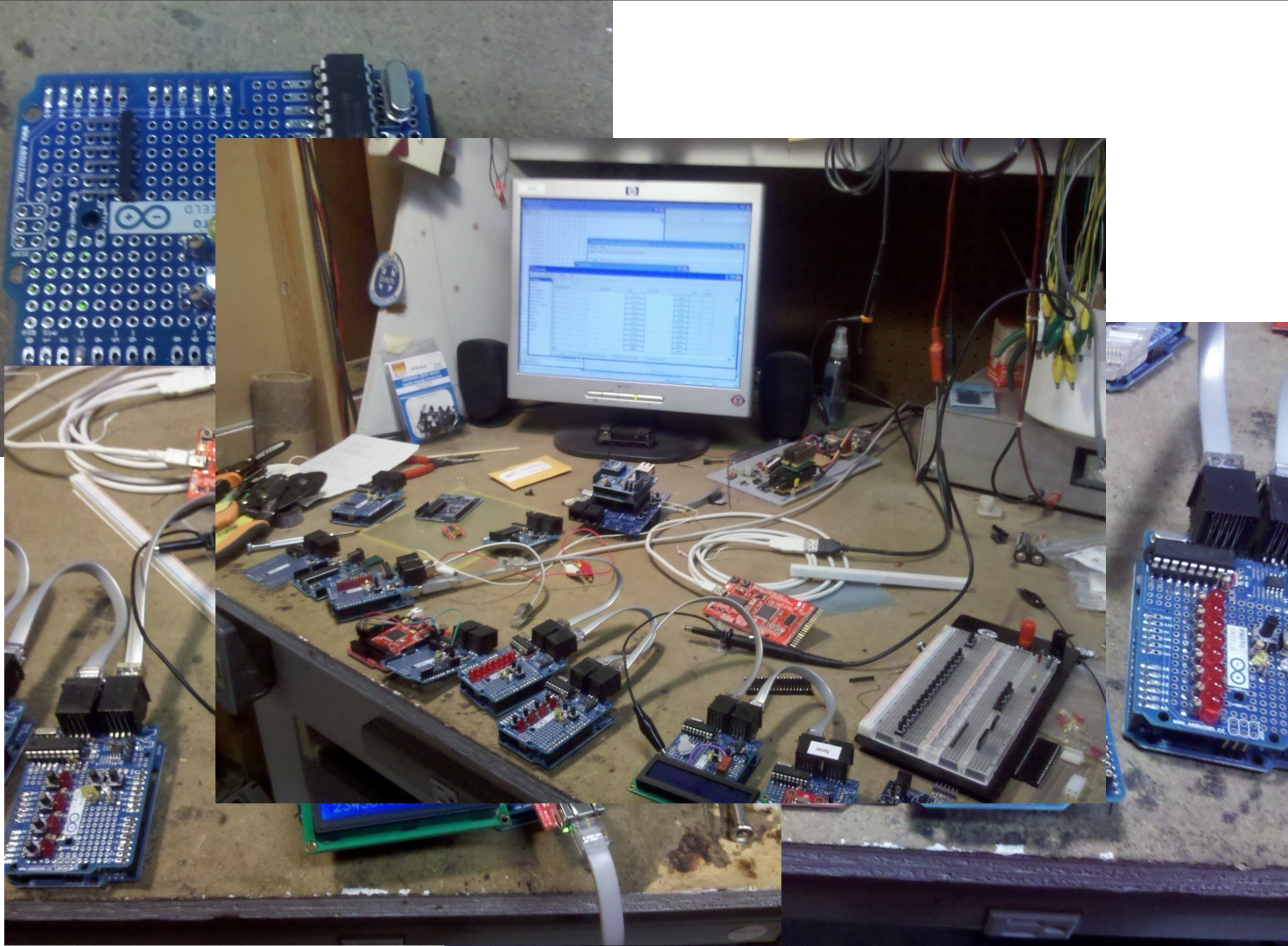
In the past year, we've put out hardware and software designs that people are using. This is Tim Hatch's work for his layout, he's also designed some boards. As John Plocher said in an earlier clinic, we can now get boards made.



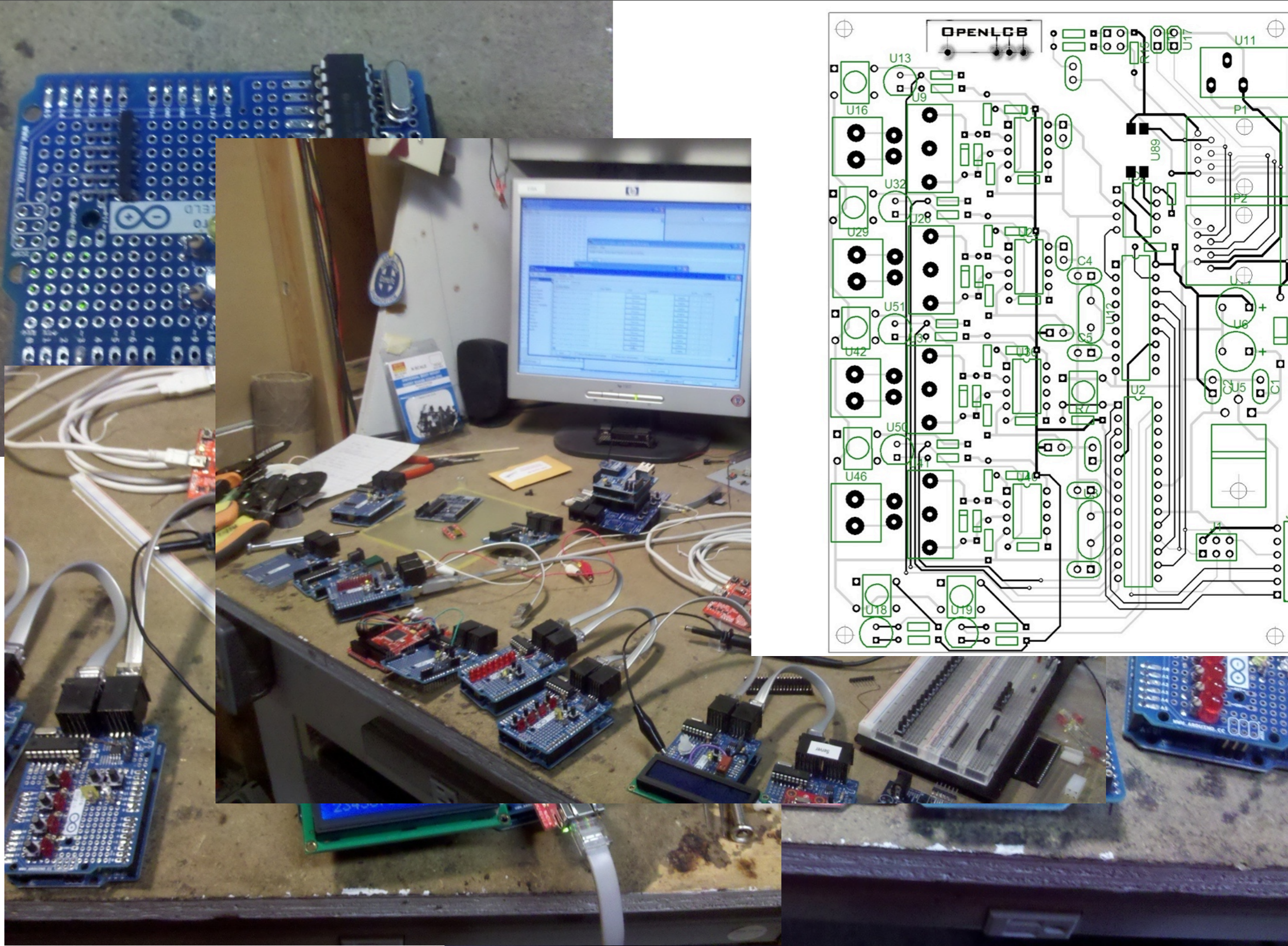
In the past year, we've put out hardware and software designs that people are using. This is Tim Hatch's work for his layout, he's also designed some boards. As John Plocher said in an earlier clinic, we can now get boards made.



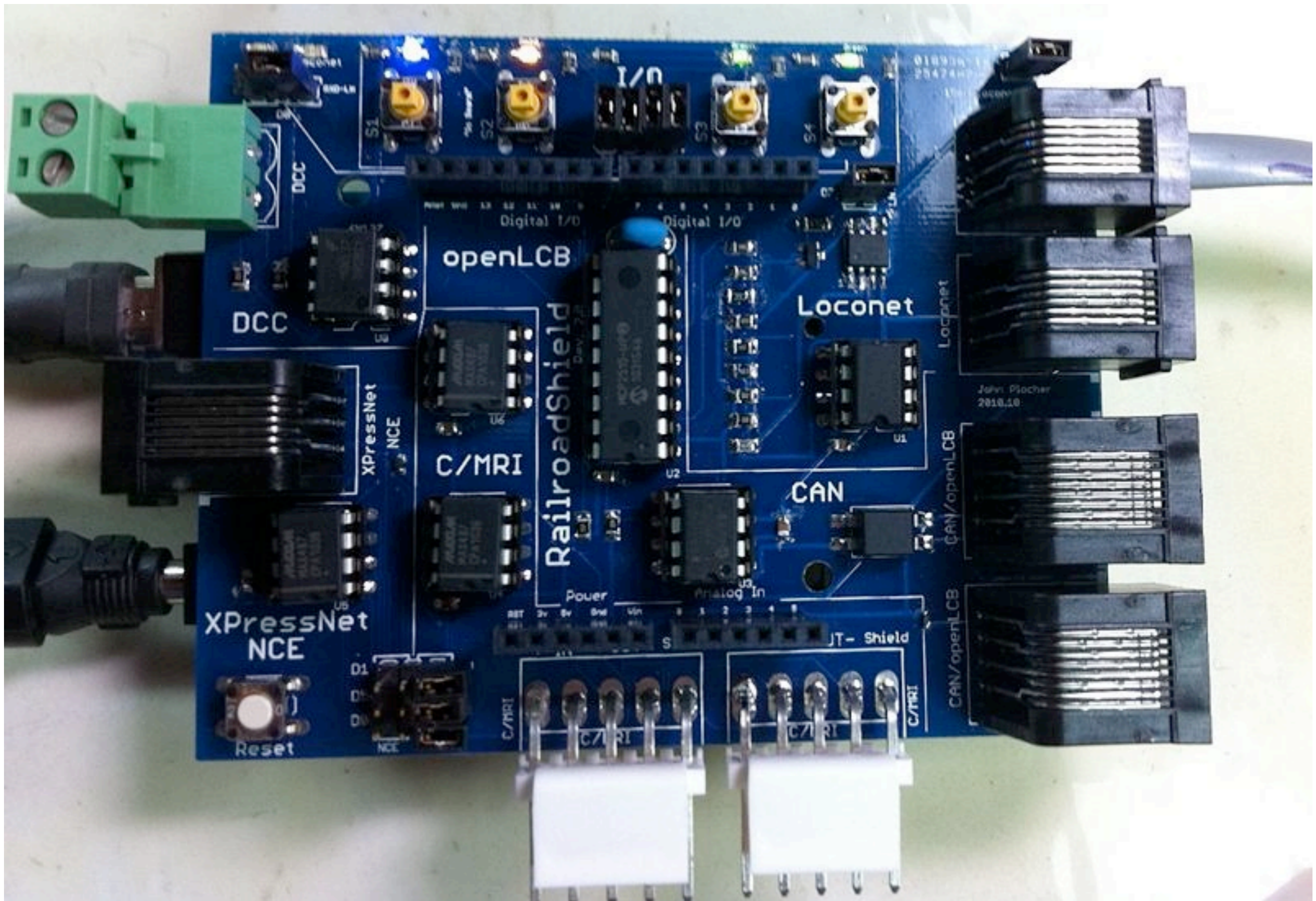
In the past year, we've put out hardware and software designs that people are using. This is Tim Hatch's work for his layout, he's also designed some boards. As John Plocher said in an earlier clinic, we can now get boards made.



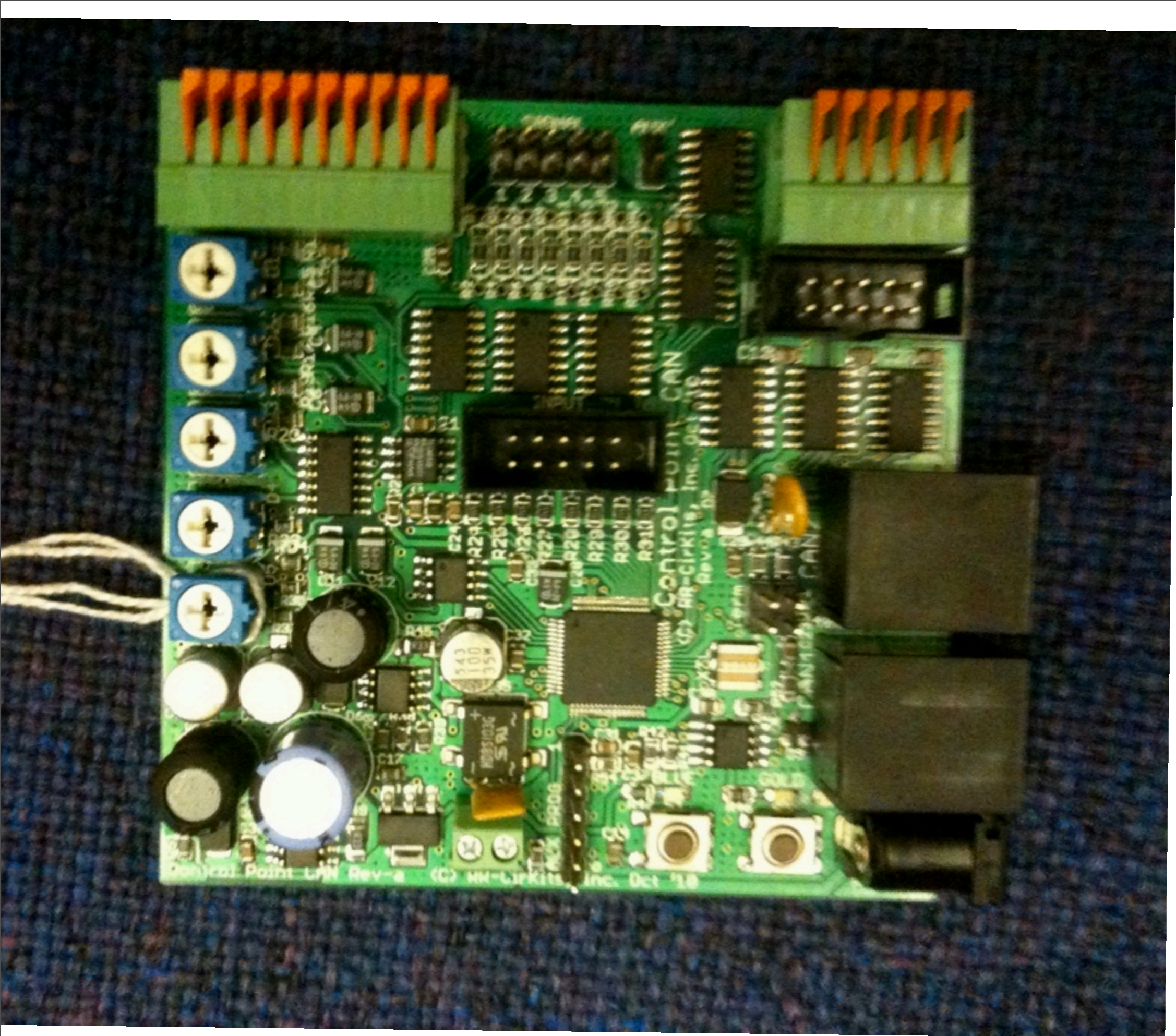
In the past year, we've put out hardware and software designs that people are using. This is Tim Hatch's work for his layout, he's also designed some boards. As John Plocher said in an earlier clinic, we can now get boards made.



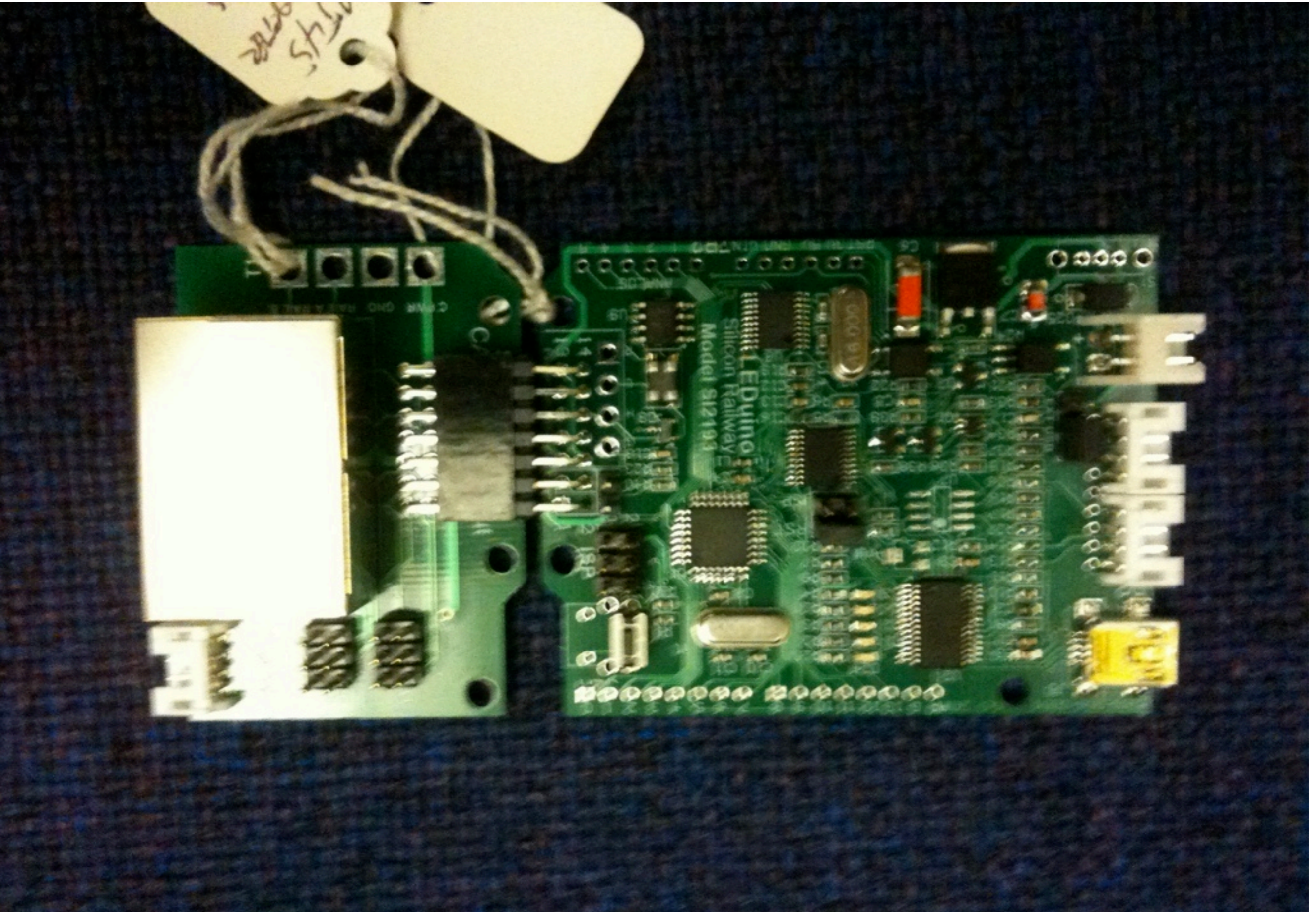
In the past year, we've put out hardware and software designs that people are using. This is Tim Hatch's work for his layout, he's also designed some boards. As John Plocher said in an earlier clinic, we can now get boards made.



Time to start talking about cables....



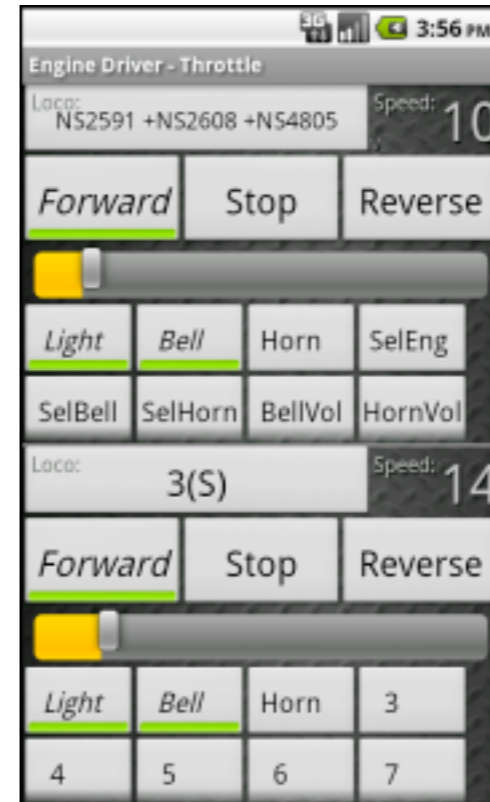
(Dick Bronson board)



(Now do the demo)

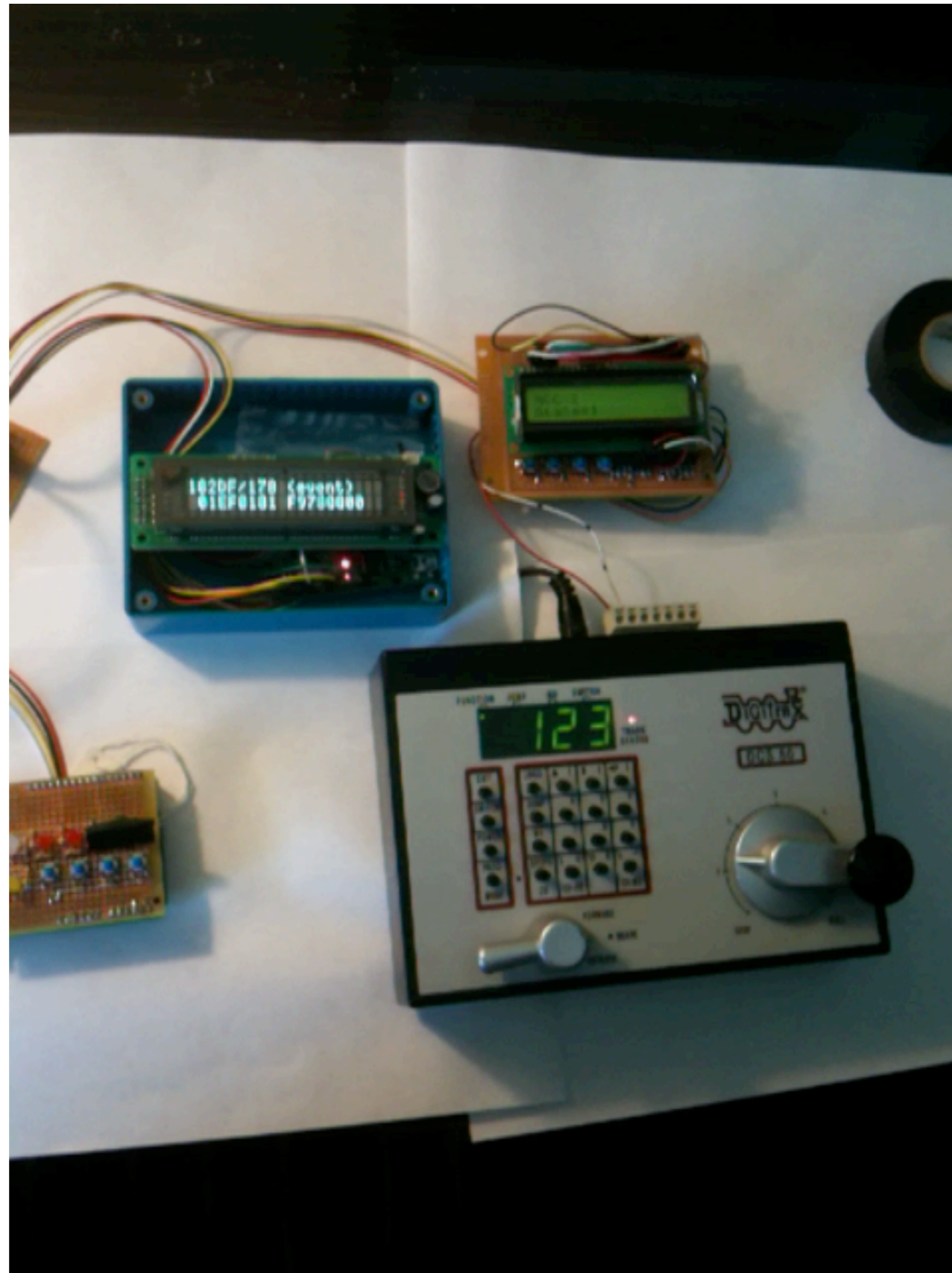


http://www.youtube.com/watch?v=XspEvkrDgls&feature=player_detailpage
Don Goodman-Wilson – the next generation of model railroaders and companies. Not everything is an event. This uses more powerful (complex to engineers) mechanisms called datagrams to move data from a specific throttle to/from the command station.



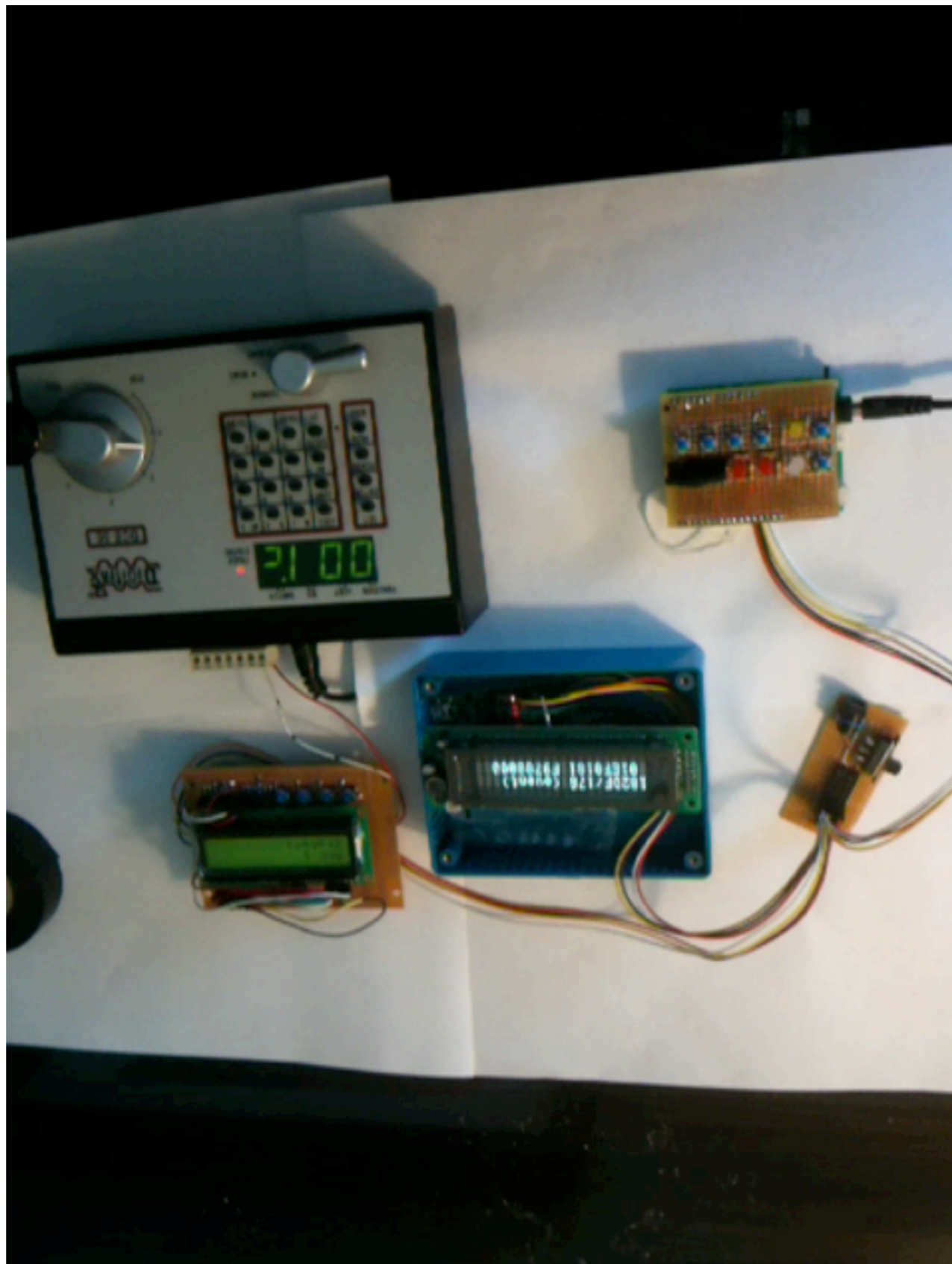
Tie this back to multiple fabrics

(do demo now)



What if you've got lots of old stuff? E.g. want to slowly add a little. Let's take example: DCC to OLCB. E.g. your NCE or Digitrax throttle, you want something to happen when you tell turnout 3 to change. **THOSE BIG MESSAGES**, but you don't have to know about them.

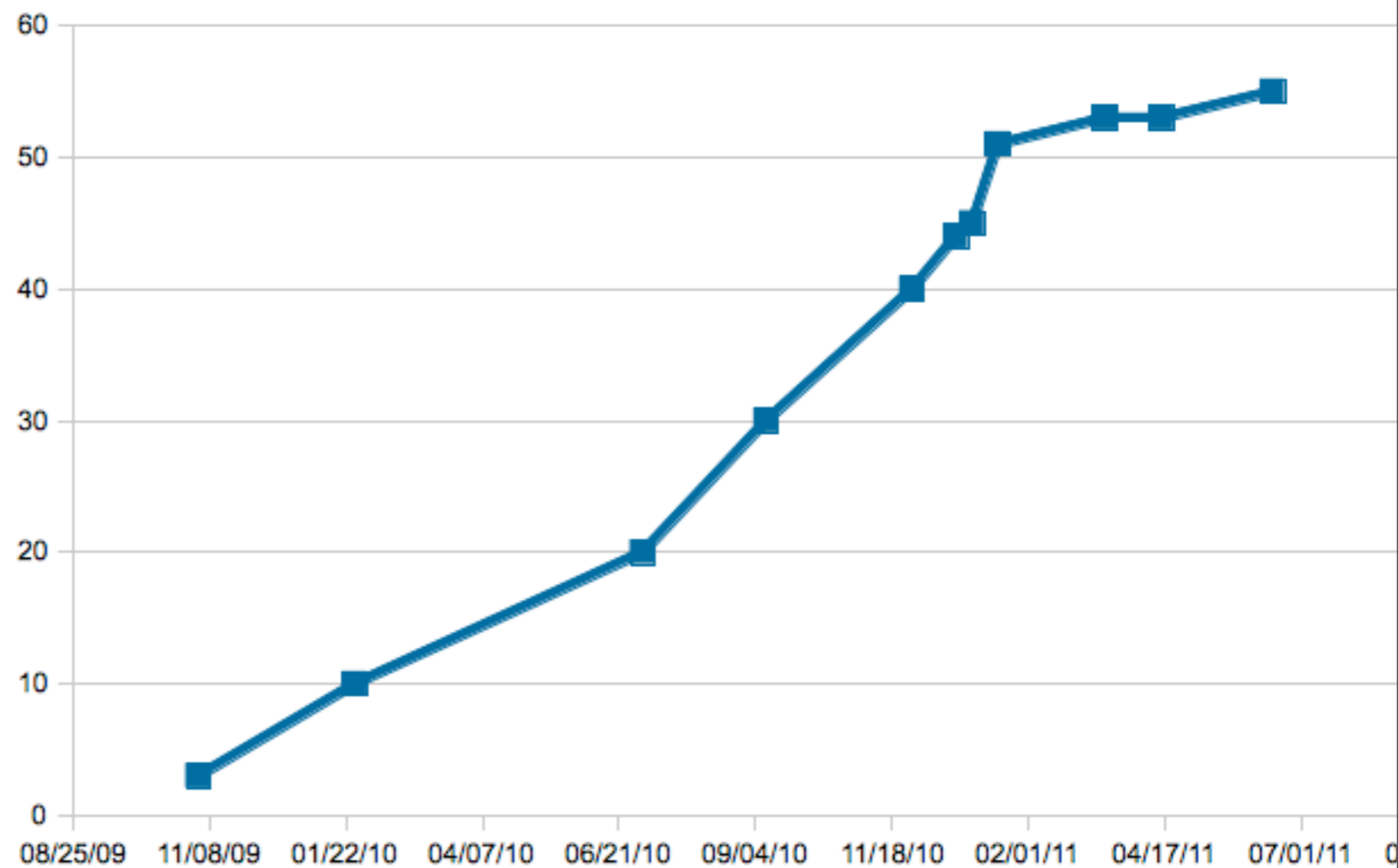
(do demo now)



Who we are and what we're doing

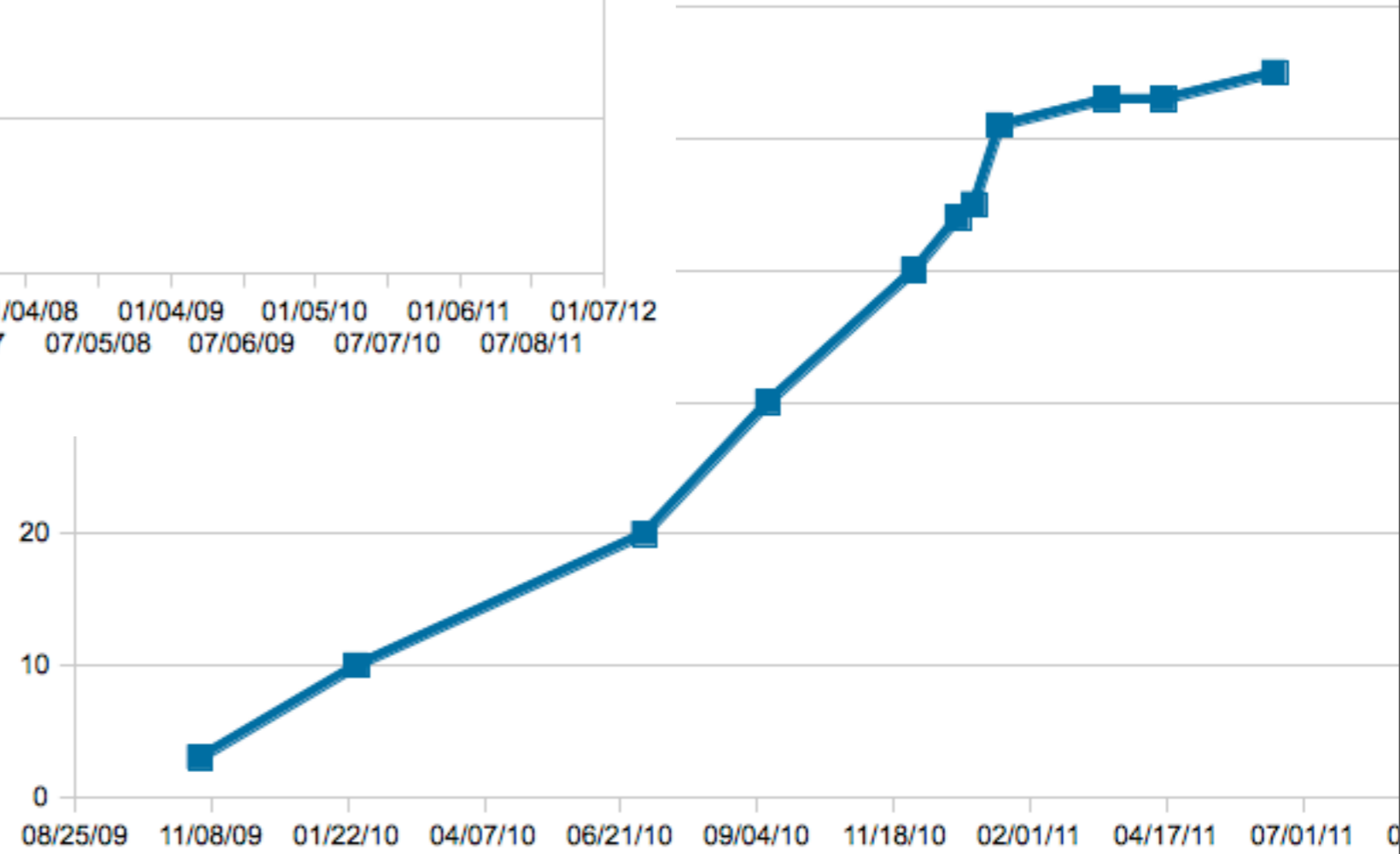
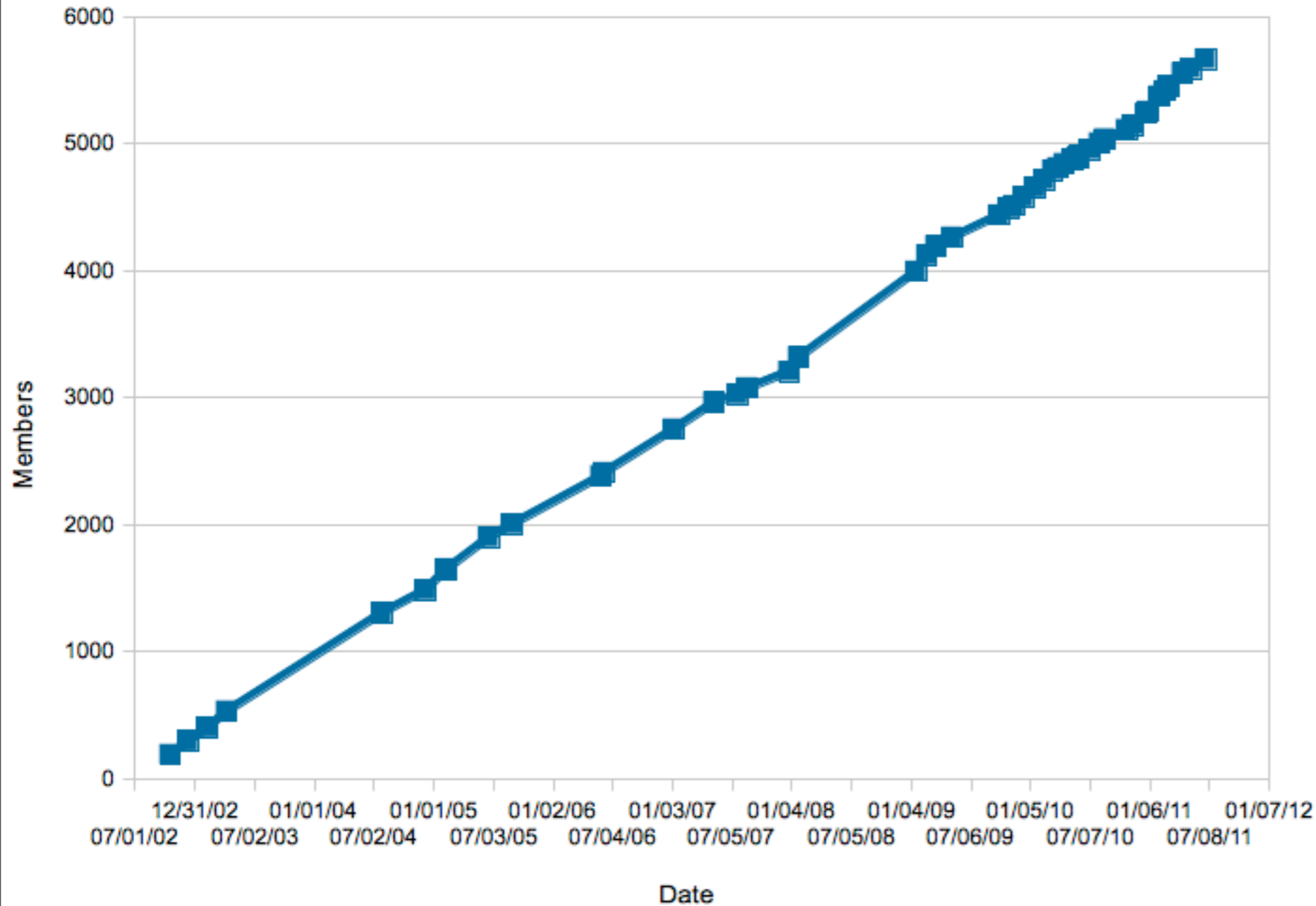
So who are we and where are we going. I'd put up pictures, but there are people working on this all over the world, and I haven't met even half of them. Like JMRI, it's a pick up group that is going great stuff.

Growth of developers list. Not users list, although not everybody on it is a developer. Drop in rate occurs due to the controversy from 9.x.1, which has caused a problem for everybody. But we've seen things like this before, and succeeded. Right on track for an effort as successful as JMRI.



Growth of developers list. Not users list, although not everybody on it is a developer. Drop in rate occurs due to the controversy from 9.x.1, which has caused a problem for everybody. But we've seen things like this before, and succeeded. Right on track for an effort as successful as JMRI.

JMRIusers membership



Growth of developers list. Not users list, although not everybody on it is a developer. Drop in rate occurs due to the controversy from 9.x.1, which has caused a problem for everybody. But we've seen things like this before, and succeeded. Right on track for an effort as successful as JMRI.

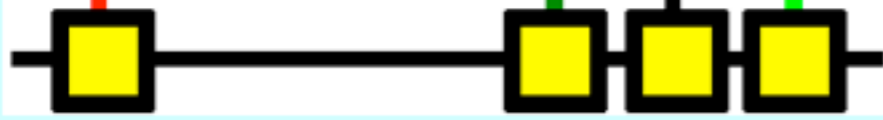
Throttle
being p

d:

- ▶ Atlas
- ▶ Broadway Limited Imports, LLC
- ▶ Bachmann Trains
- ▶ CML Systems
- ▼ CT Elektronik
 - ▶ Function Decoder
 - ▼ DCX Series (v.66 and higher)
 - ▶ DCX74D
 - ▶ DCX74DV
 - ▶ DCX74z
 - ▶ DCX75
 - ▶ DCX Series (v.27 and higher)
 - ▶ DCX Series (old)
 - ▶ Sound Decoder
 - ▶ Sound Decoder SL (v.26 up)
 - ▶ Sound Decoder SL 51
 - ▶ Sound Decoder SL (v.40 up)
- ▶ CVP Products
- ▶ DCC Concepts
- ▶ Digitrax
- ▶ Electronic Solutions Ulm GmbH
- ▶ Fleischmann
- ▶ GER Designs

So what are we working on? This is DecoderPro. I really want to get out of this business.

OpenLCB



A new way of controlling model railroads

The OpenLCB™ project is developing an easy but sophisticated network for model railroad control using today's technology.

This website describes a developing Local Control Bus which is used to control MRR accessories. It is designed to make the set-up of control systems, be they simple or complex, much easier, and to provide complex control arrangements very simply. LCB's do not yet control the actual trains, we don't propose to replace DCC. It adds the ability for layout features and your operator consoles or CTCs to more effectively control the trains and the layout by bringing everything together in one system. Your DCC investment is protected, as is your CTC investment, and your signals and turnouts. What an LCB does is allow you to bring these things together. Cabling and connection should be simple, low cost, and probably pre-made but can be DIY assembled if needs be. No control wiring should be needed, only a single network cable and a single power supply cable. Configuration should be simple and easy to manage. To see how this works, look at the second video on [this](#) page - which shows how things are actually programmed. The third video shows some more-complex programming. You should be able to re-use things you have, like signals, BOD's, turnouts, train ID sensors, RFID, structure lighting and the like, but connect them in ways you never thought possible to achieve full prototypical accuracy and detail. Add complex new technology easily, such as sound effects, TV in structures, timetable signs in stations, outdoor video displays, structure lighting fully controlled and much more as time goes on. Hopefully different manufacturers will support a common standard and so the range of products available will grow.

From small ... to big ... to huge,

OpenLCB™ will help you do what you want to do most: run your railroad.

It is Easy to use: Whether you want to do simple setup by just pushing buttons on boards, or use a full-featured configuration tool, OpenLCB makes setup easy.

It protects your investment: From your first two boards through huge modular layouts, OpenLCB equipment will grow with your railroad, and you can still connect and use your legacy equipment.

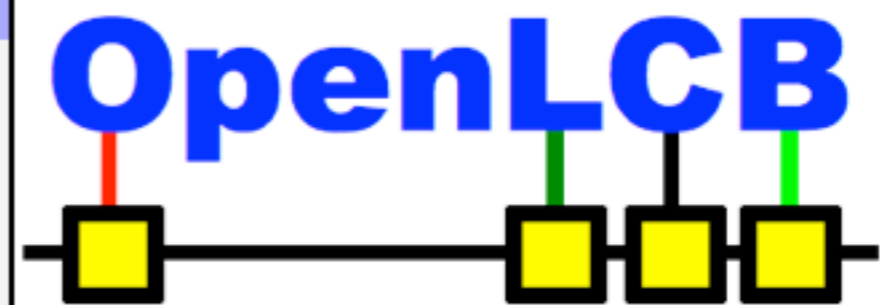
It is simple to expand: OpenLCB equipment can be added without worrying about ID conflicts, without keeping track of complicated address assignments, and never being forced to go back and reconfigure equipment - all while working with your existing model-railroad controls.

People working on documentation for e.g. user level web pages (even took some grief about spending time on a new logo, but part of community efforts is allowing lots of types of people to contribute)

OpenLCB Specifications

This page provides access to the OpenLCB™ Standards and Technical Note documents.

OpenLCB is not a formal standards-making body, but we are interested in making it clear what's required and not required for interoperation between OpenLCB nodes. Therefore, we are creating a series of "standards" and corresponding explanatory "technical notes" as specific documents. "Standards" are normative, to the extent that anything this bunch of happy campers does can be proscriptive. "Technical Notes" are coupled statements of explanations and additional information that go with each standard. For more information on the conventions used when writing OpenLCB standards and technical notes, please see the separate "[conventions](#)" page.



For background information, informal and development documents, etc, please see the "[documentation](#)" index page.

Layered View

OpenLCB has been created as a layered family of protocols. The basic structure is along the lines of the wildly successful architecture of the ARPA TCP/IP family of protocols. For use on e.g. Controller Area Network (CAN), OpenLCB has to deal with lower-level concepts than the TCP/IP stack does, however, so OpenLCB is structured like a hybrid of the TCP/IP with the lower-level physical-layer and data-link-layer concepts of the ISO model.

In the following, items marked in **light green** are complete, at least for a first pass. Items marked in **blue8** are preliminary and under development. Uncolored links are to informal documents which will eventually form adopted standards and technical notes. The right two columns list the corresponding ISO and TCP/IP layer names, and are not part of OpenLCB.

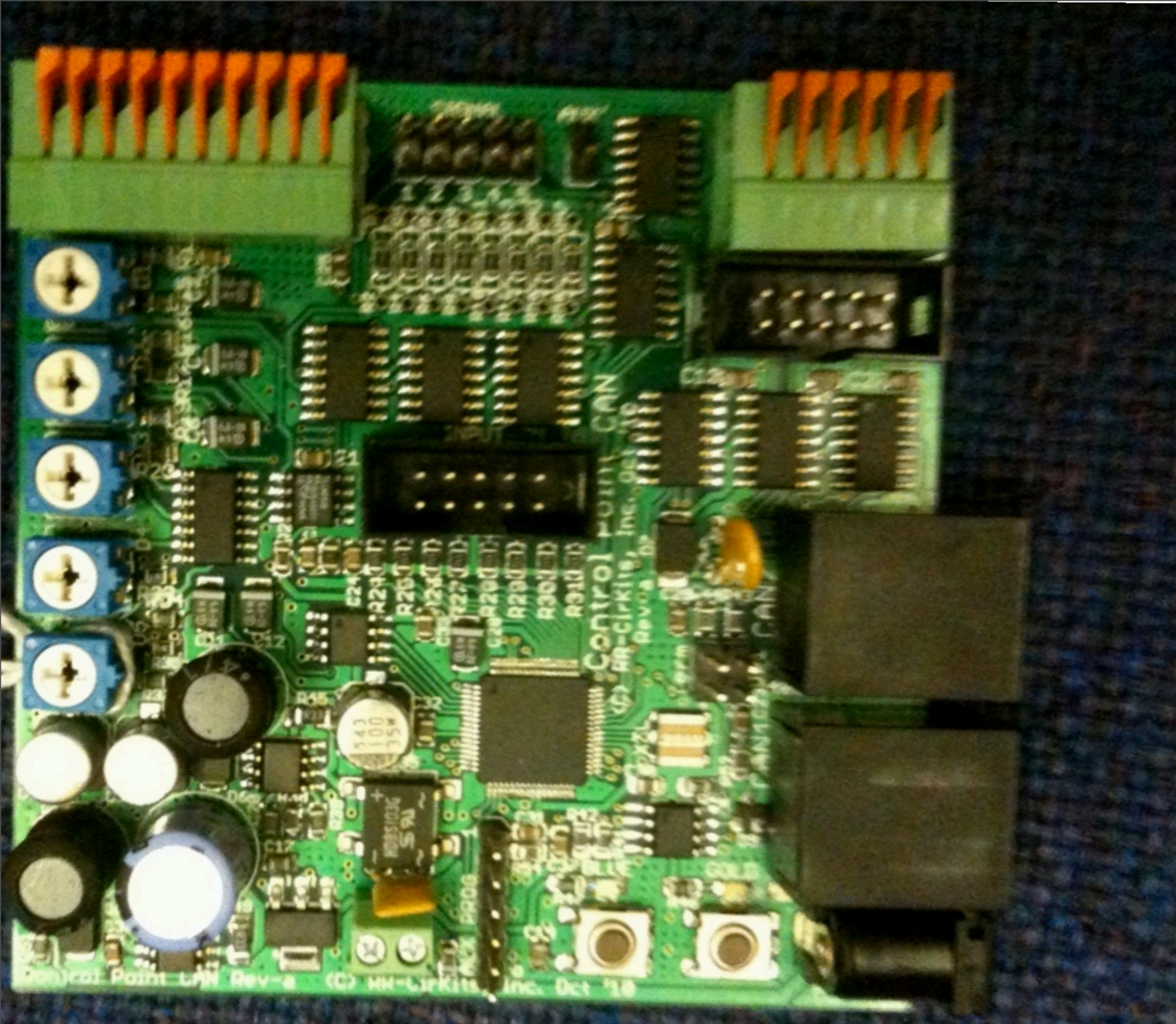
Introduction (We need a strong technical, non-normative introduction of some sort. It's not clear where it goes in this view of the table of contents. See the prior "[Technical Introduction](#)" [development document](#) for an example of possible content.)

Common	CAN	TCP/IP	ISO Model Layer	TCP/IP Model Layer
Display Protocol [dev note]			<i>Application</i>	<i>Application</i>
Configuration Description Information [dev note]				
Configuration Protocol [dev note]				
Remote Button Protocol [dev note]				
Ident Method [dev note]				
Event Teaching/Learning Protocol [dev note 1] [dev note 2] [dev note 3]				

Doing lots of writing. It makes your thoughts more precise. "A specification that nobody uses is just a particularly irrelevant form of science fiction" This lets us have parallel implementations that show that things really work. 5 right now (C for Arduino; separate C for PIC; C++ for AVR; Java for JMRI; Objective C for iOS)

Producer/Consumer Protocol [dev note]				
Protocol Identification Protocol [prelim Std (.pdf)(.odt)] [prelim TN (.pdf)(.odt)]				
			<i>Presentation</i>	
			<i>Session</i>	
	OpenLCB-CAN Stream Transport [dev note] OpenLCB-CAN Datagram Transport [prelim Std (.pdf)(.odt)] [prelim TN (.pdf)(.odt)] [dev note] OpenLCB-CAN Event Transport [prelim Std (.pdf)(.odt)] [prelim TN (.pdf)(.odt)] [dev note]	OpenLCB-TCP/IP Stream Transport OpenLCB-TCP/IP Datagram Transport OpenLCB-TCP/IP Event Transport	<i>Transport</i>	<i>Transport</i>
	OpenLCB-CAN Message Network [prelim Std (.pdf)(.odt)] [prelim TN (.pdf)(.odt)] [Standard Interactions]	TCP/IP Message Network [Standard Interactions]	<i>Network</i>	<i>Internet/Network</i>
Unique Identifiers (Node IDs) [prelim Std (.pdf)(.odt)] [prelim TN (.pdf)(.odt)] Event Identifiers [prelim Std (.pdf)(.odt)] [prelim TN (.pdf)(.odt)]	OpenLCB-CAN Frame Transport [prelim Std (.pdf)(.odt)] [prelim TN (.pdf)(.odt)]	OpenLCB-TCP Segment Transport [prelim Std (.pdf)(.odt)] [prelim TN (.pdf)(.odt)] [Standard Interactions] [TCP/IP (binary protocol) details]	<i>Data Link</i>	<i>Network Interface/Subnetwork</i>
	OpenLCB-CAN Physical Layer [Standard] [Technical Note] Also submitted to NMRA (S 9.x.1) (TN 9.x.1)	(none)	<i>Physical Interface</i>	

Here's your chance to talk about NMRAnet if you want.



So how can you get involved? (pictures of RR-circuits, LEDUino, throttle) Intended commercial products (have cards out, intro all three) But we want you to come play. Lesson of JMRI: The more cooks, the more eats.

How to get in touch

- <http://openlcb.org>
- <http://groups.yahoo.com/group/openlcb>

Extra Slides

