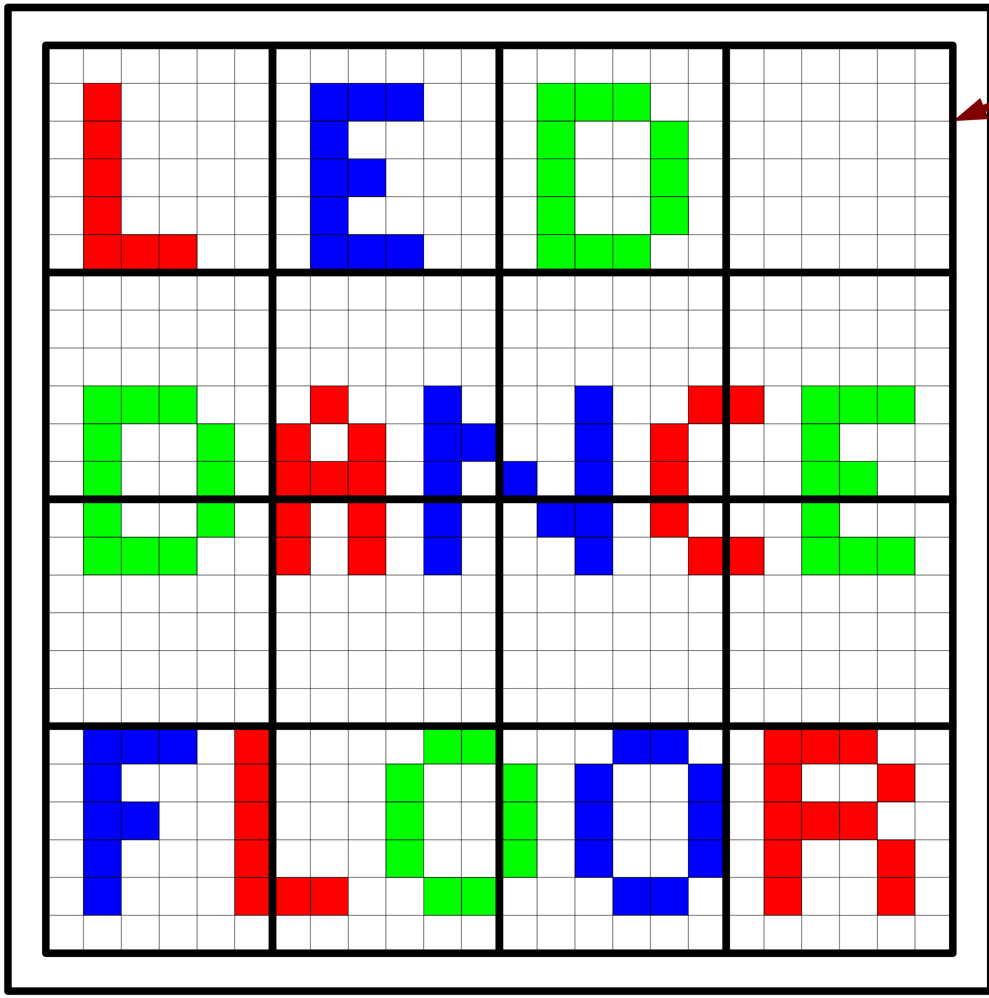
LED dance floor, by Nick, Kathryn, Mark, and Aiden

The inspiration – a game floor at Kidzone

$23,000!

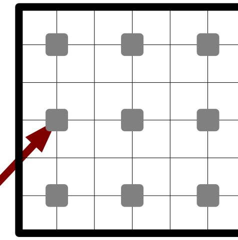
OUCH!

I think we can beat that!



8'x8' Dance floor is made of 4'x4' quarter-floor modules...

... and a 4in border for control electronics and cabling



Each quarter-floor module is 12x12 cells, sized to fit the acrylic we could find.
Each cell is 4x4in, minus “1in” (before planing) spruce which makes up the cell walls, slotted into each other.
There are pressure-switches in between each 2x2 cells to detect players standing/jumping there (so 12x12 sensors total).

Side view of a cell:

Pressure-switch

Acrylic, frosted

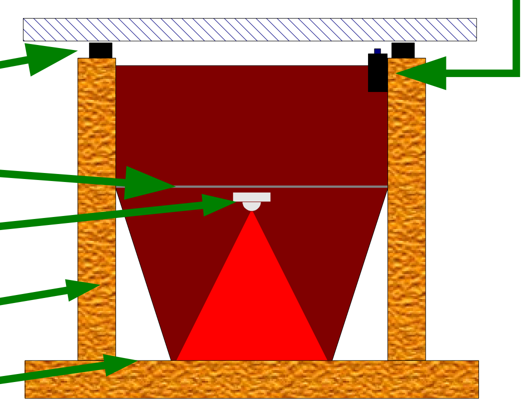
Neoprene draft excluder / weatherstripping for “bounce”

Wiring – see later

RGB LED

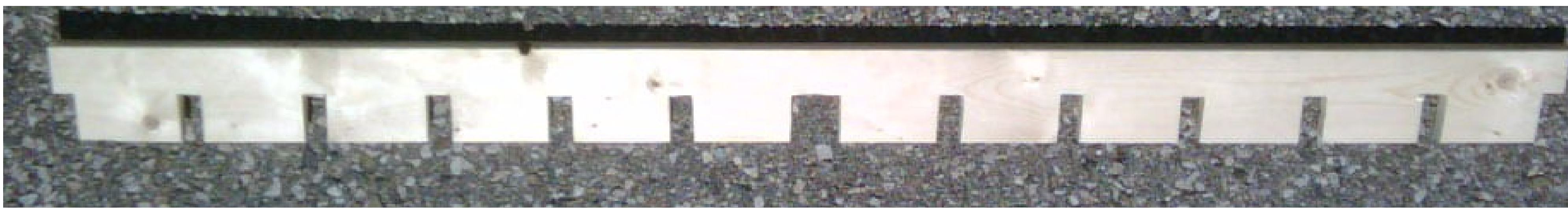
Cheap spruce boxing, painted white

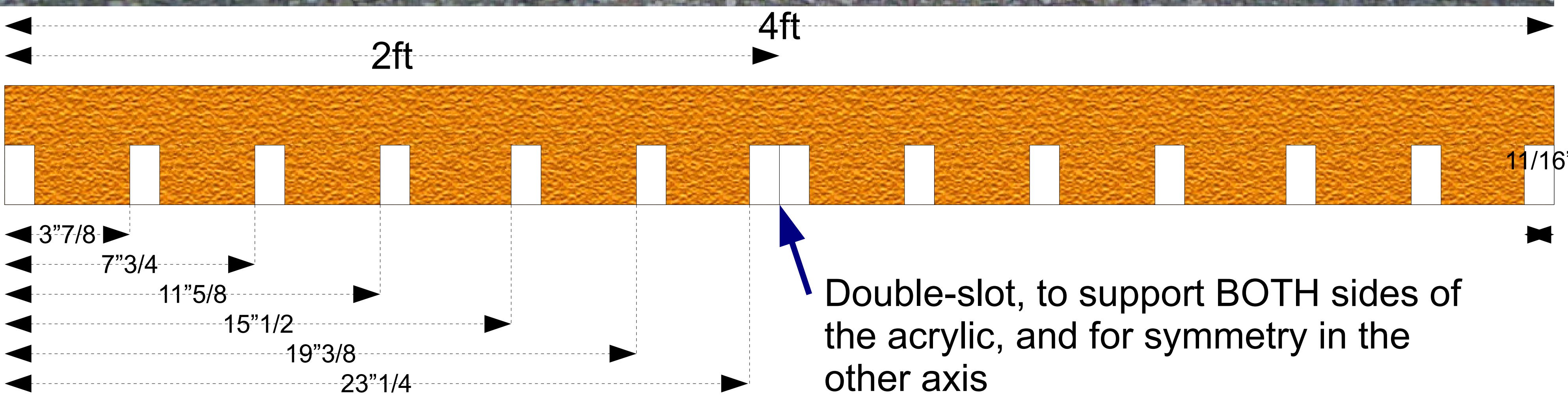
Base board, painted white



Boxing - Wood cutting guide

After planing, 1”x4”x8' wood is more like 11/16” x 3”7/16 x 8'
So slots for boxing need to be 11/16” wide and 1”3/4 deep.
All 4 sides of the 2'x4' acrylic need proper support.
We're making slightly-fake 2'x2' boxes, so one sheet of acrylic sits over 2 such boxes.
We're building the dance-floor in four 4'x4' quarters for a certain amount of portability. Each quarter will hold 2 sheets of 2'x4' acrylic, with wood doubled where required to support the edges of the acrylic and for symmetry



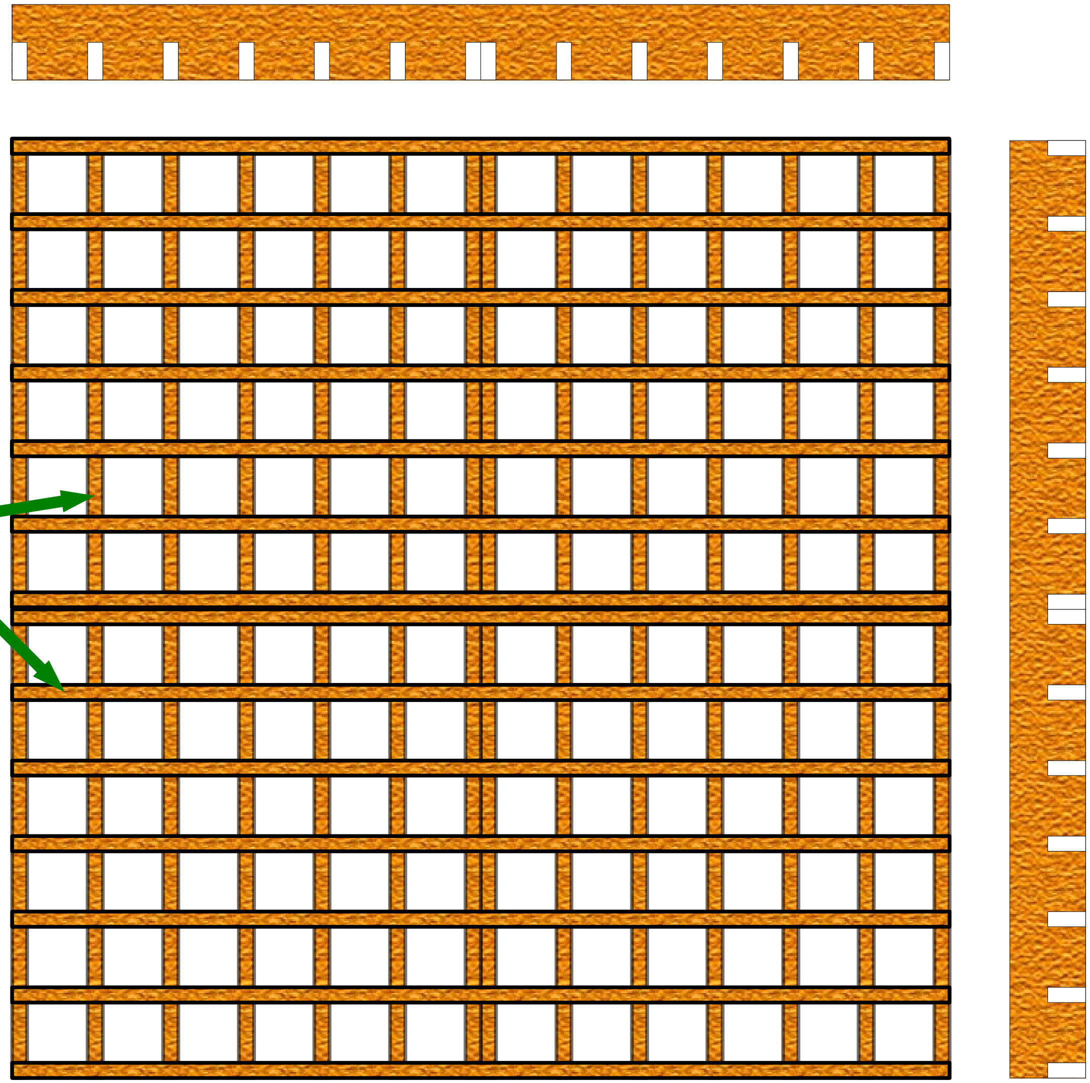


To my Brit friends, I apologise. Canada pretends to be metric, they use litres, km, kph, etc, but all the tools, acrylic, and wood come in bloody imperial sizes! Grrrr

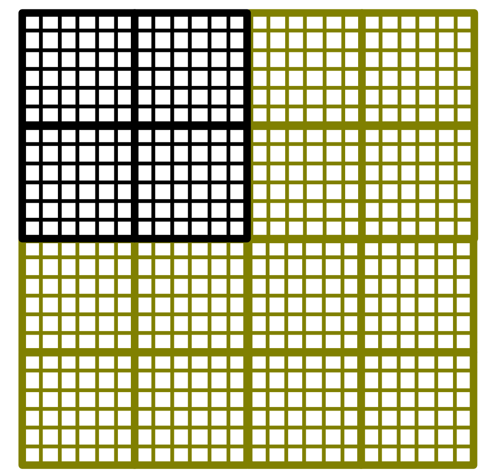
Main floor assembly - boxing

This is a quarter-floor.

It uses a total of 28 identically-cut pieces of wood, (14 N-S under-neath and 14 E-W slotting into them from above)



2x2 of these units make the entire floor:

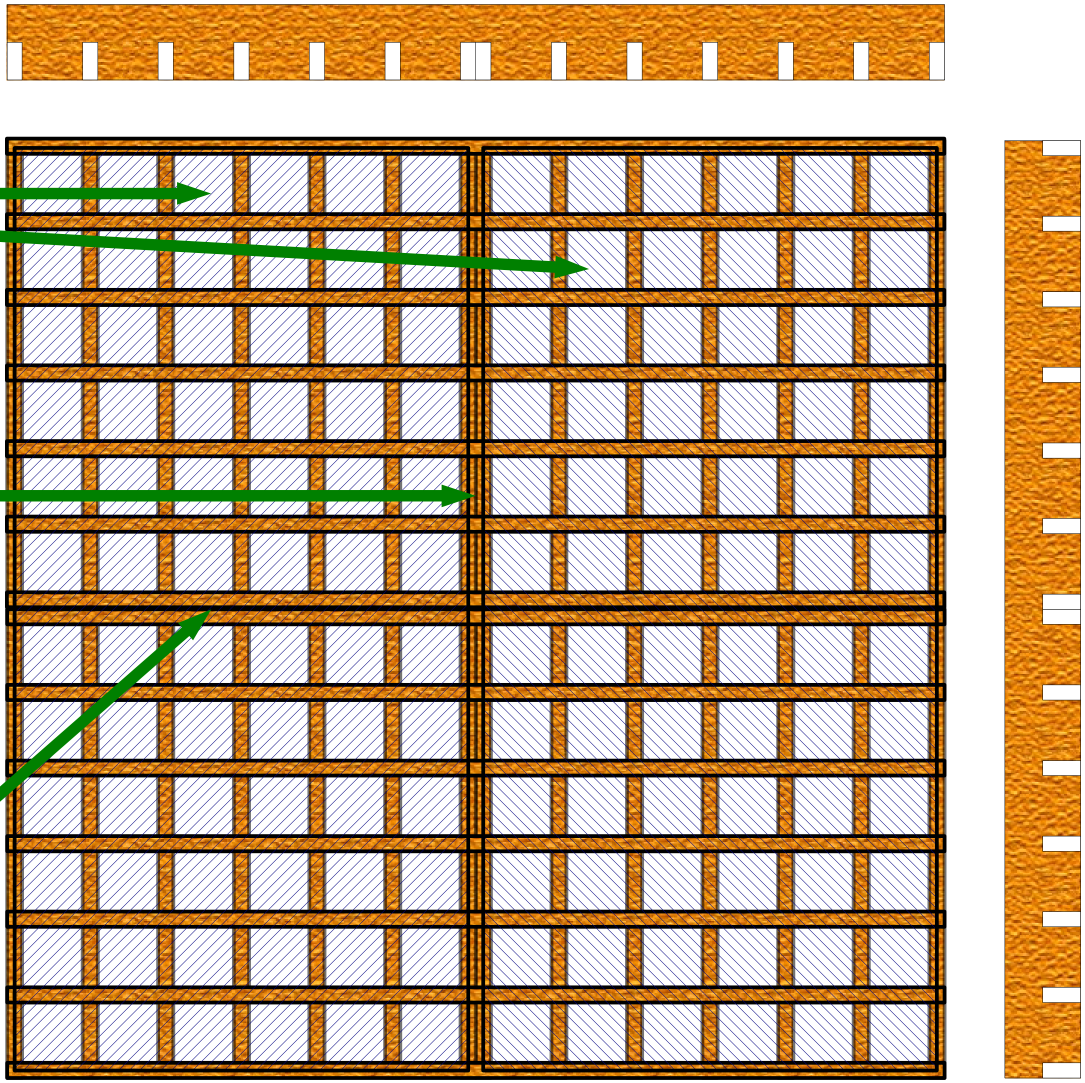


Main floor assembly - acrylic

2 sheets of 2'x4' frosted acrylic

The doubled-up verticals are there to nicely support the edges of the acrylic

The doubled-up horizontals are not structural, just there for symmetry ☺

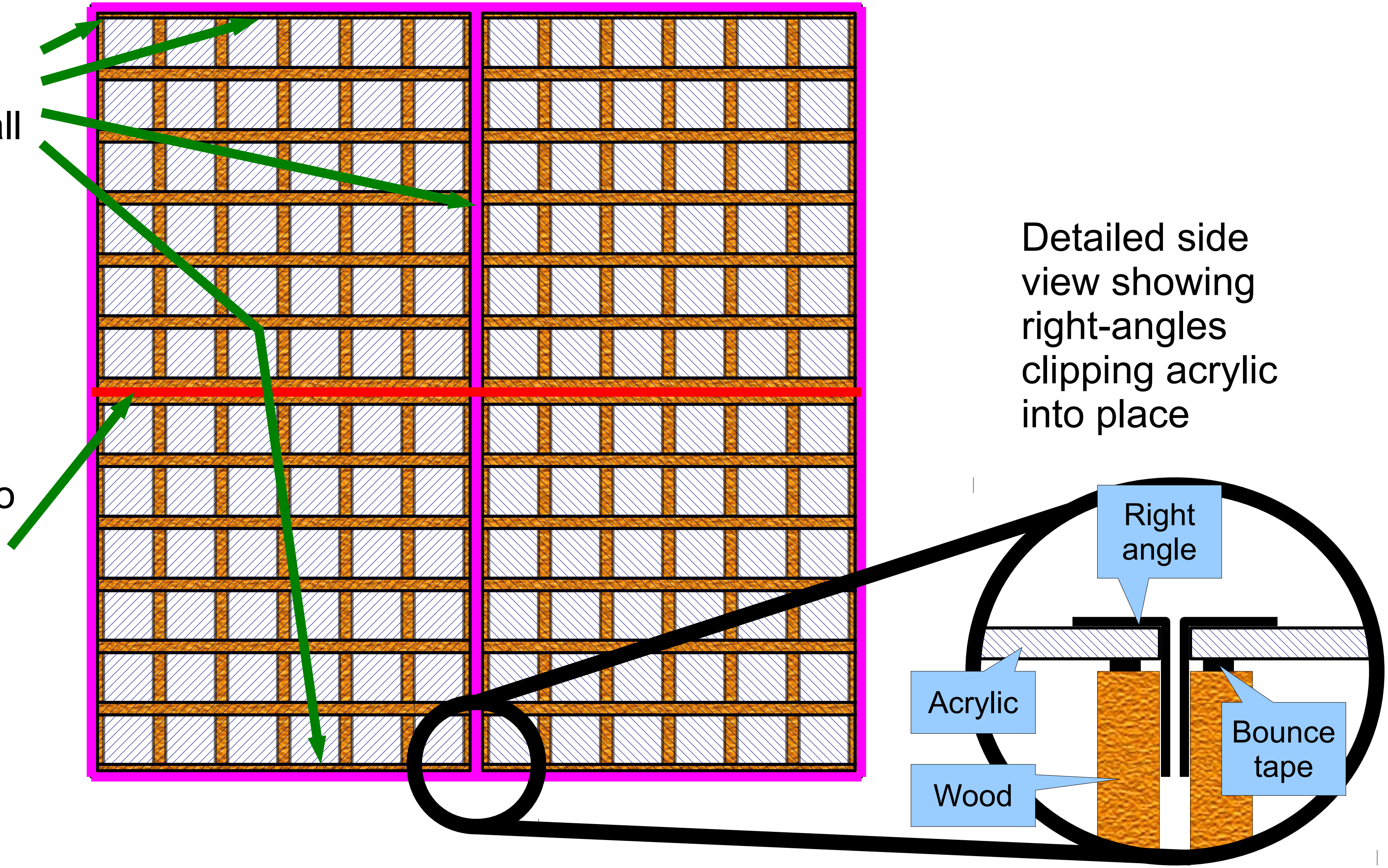


Main floor assembly - right-angles

Right-angles clip the acrylic into place on all 4 edges of each sheet

Dummy tape to look like right-angles, for symmetry

Detailed side view showing right-angles clipping acrylic into place



LED Dance Floor v1+v2 Wiring

- Our previous Dance Floor design was going to use Piranha ultra-bright RGB LEDs
- We ran 3 wires down each column (R,G,B), one wire across each row (common anode)
- We had a complicated collection of shift-registers to power 1 row whilst grounding the corresponding RGB columns to display the row contents
- Whilst displaying 1 row, we were clocking the next row's contents into the shift registers
- If we displayed each row in quick succession, persistence-of-vision makes us see the entire display

  [See end of slide deck if you want details]

LED Dance Floor v1+v2 Wiring

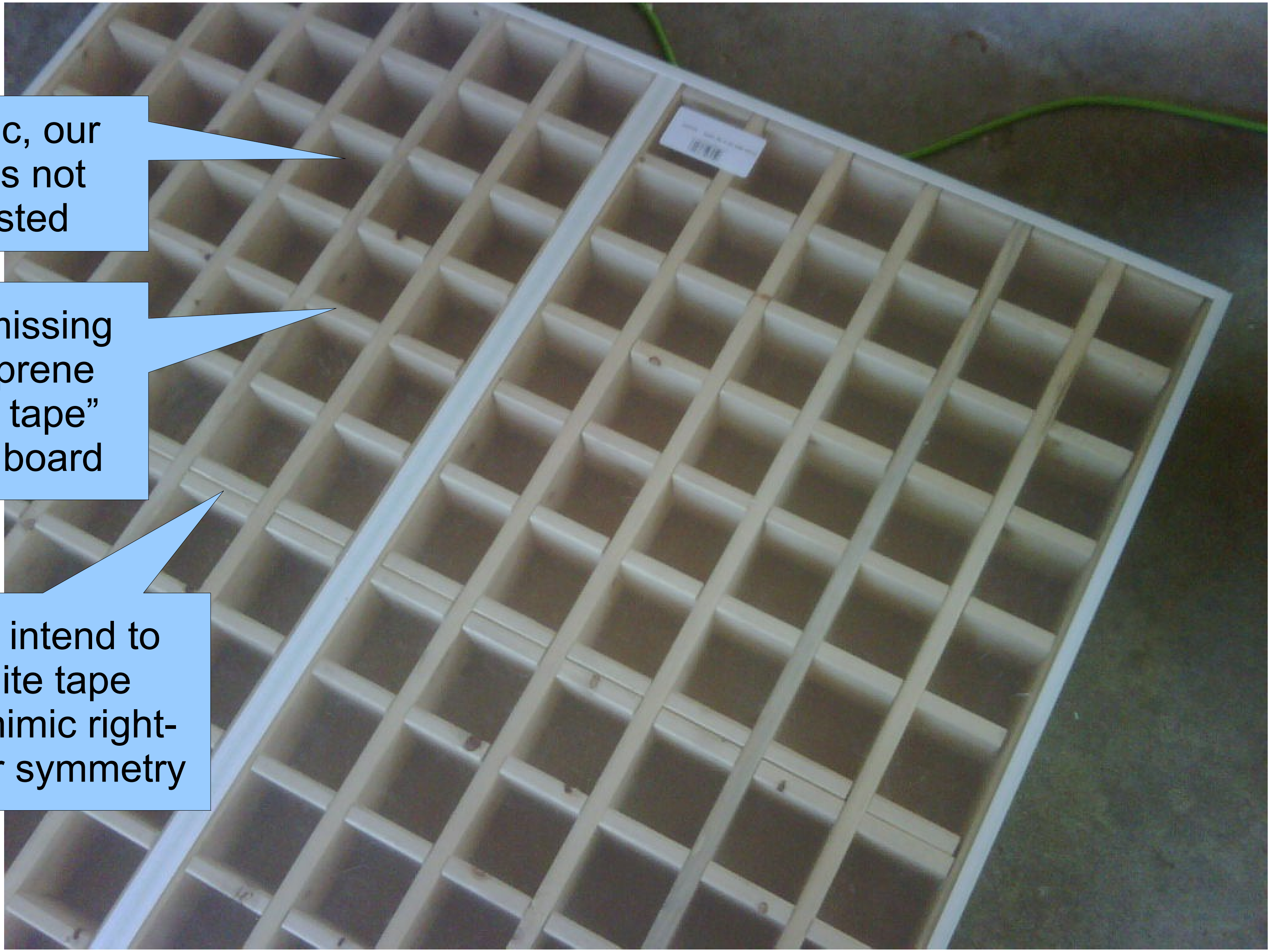
- Version 1 had the shift-registers controlled by a parallel port, which was terrible, mostly because modern parallel ports “ain't what they used to be”, they buffer up blocks of data, mess with your timing, and try to “negotiate” with a printer (or a dance floor which doesn't know how to negotiate as a printer, and is confused by the extra data)
- Version 2 was very similar except an Arduino microcontroller drove the shift-registers. This worked much better, and also allowed us to code on the Arduino, unplug the laptop, and have the floor run by itself!

Testing 3x3 cells

We first built a 3x3-cell test mini-floor, to experiment with different sized cells, test the strength of the acrylic, experiment with different sensors, and to test the electronics (including the shift-registers etc) on a small scale

First Quarter Test! (no LEDs)

Main floor assembly – first quarter!



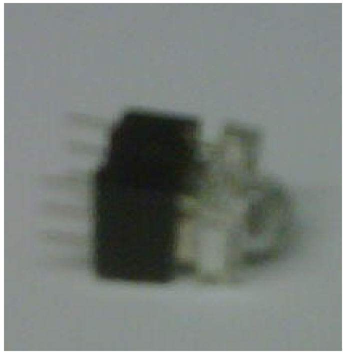
In this pic, our acrylic is not yet frosted

We are missing our neoprene “bounce tape” on each board

We also intend to put white tape here to mimic right-angles for symmetry

Problems with v1+2 Wiring

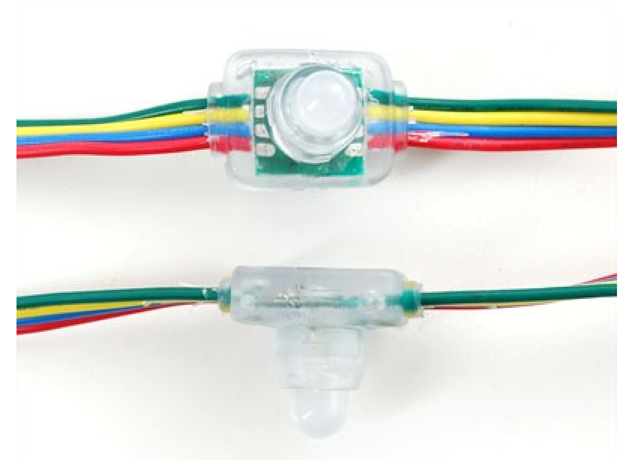
- When we scaled our 3x3-cell test up to a 12x12-cell 4'x4' quarter-floor, never mind an entire 24x24-cell floor, the huge soldering effort proved impractical
- It was too easy to burn out LEDs if we soldered direct to them, we couldn't find appropriate LED holders, but found we could make our own from chip holders

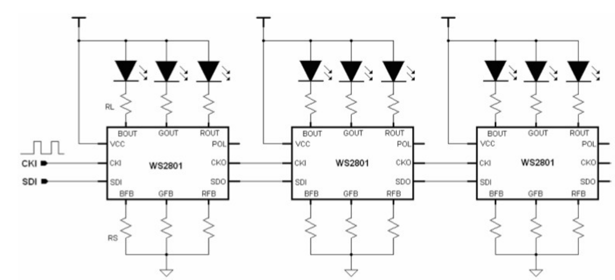


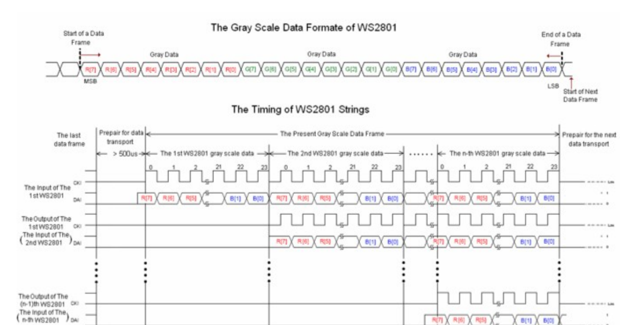
- 24x24 cells required cutting, breaking, filing, glueing 288 chip holders into 576 LED holders. We didn't complete this!
- These had to be soldered into the floor in-situ, at awkward angles.
- 4 connections per cell, very close to each other, would require over 2300 hand-soldered joints that **must be perfect**, with no open circuits, and no shorts
- It would have been **really** hard to track down and fix any shorts due to rows and columns being powered “wrong”

LED Dance Floor v3 Wiring

- ... but thankfully we discovered chains of WS2801 RGB LEDs!
- They are pre-wired with GND, +5V, clock, and data lines
- A simple serial protocol, easily driven by an Arduino (or similar), can clock 8 bits of R, 8xG, 8xB into each LED before moving onto the next LED in the chain
- Once you stop clocking for 500µS, all LEDs display their individual 24-bit colour values and you can begin again at the beginning
- The clock/data rate is sufficiently high that we can trivially manage great frame-rates even if we had much bigger dance-floors

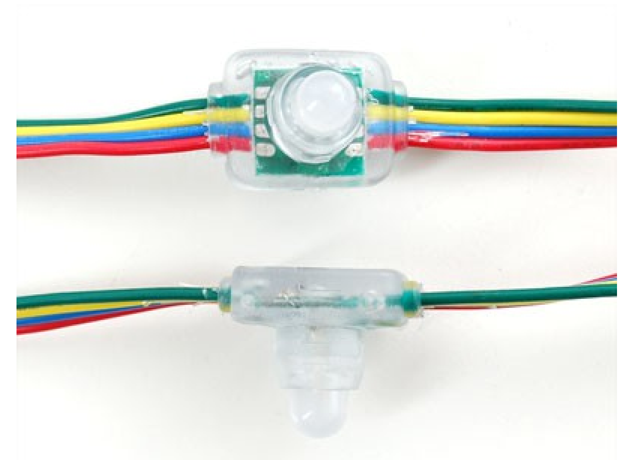


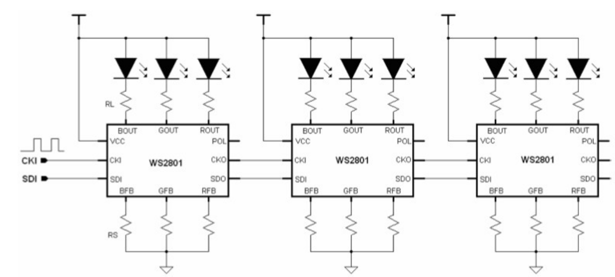


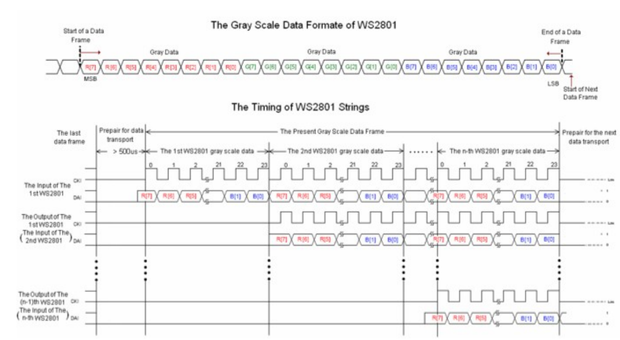


LED Dance Floor v3 Wiring

- Our Arduino **could** clock into (say) 8 chains simultaneously for even higher frame-rates
- ... but we really don't need to. 25 MHz clock-rate could update a million LEDs per second
  ... or 50,000 LEDs at 20fps
  ... or 20,000 LEDs at 50fps
  ... or our 24x24 LEDs at 950fps – PLENTY!
- These cost more, however, there's almost no soldering, no chip-holders to make, no shift-registers, resistors, transistors
- We just drop them into the floor and connect them up to the Arduino and to power!
- They are also 24-bit instead of 3-bit,
  and brighter due to illuminating ALL LEDs at the same time, not one row at a time / POV



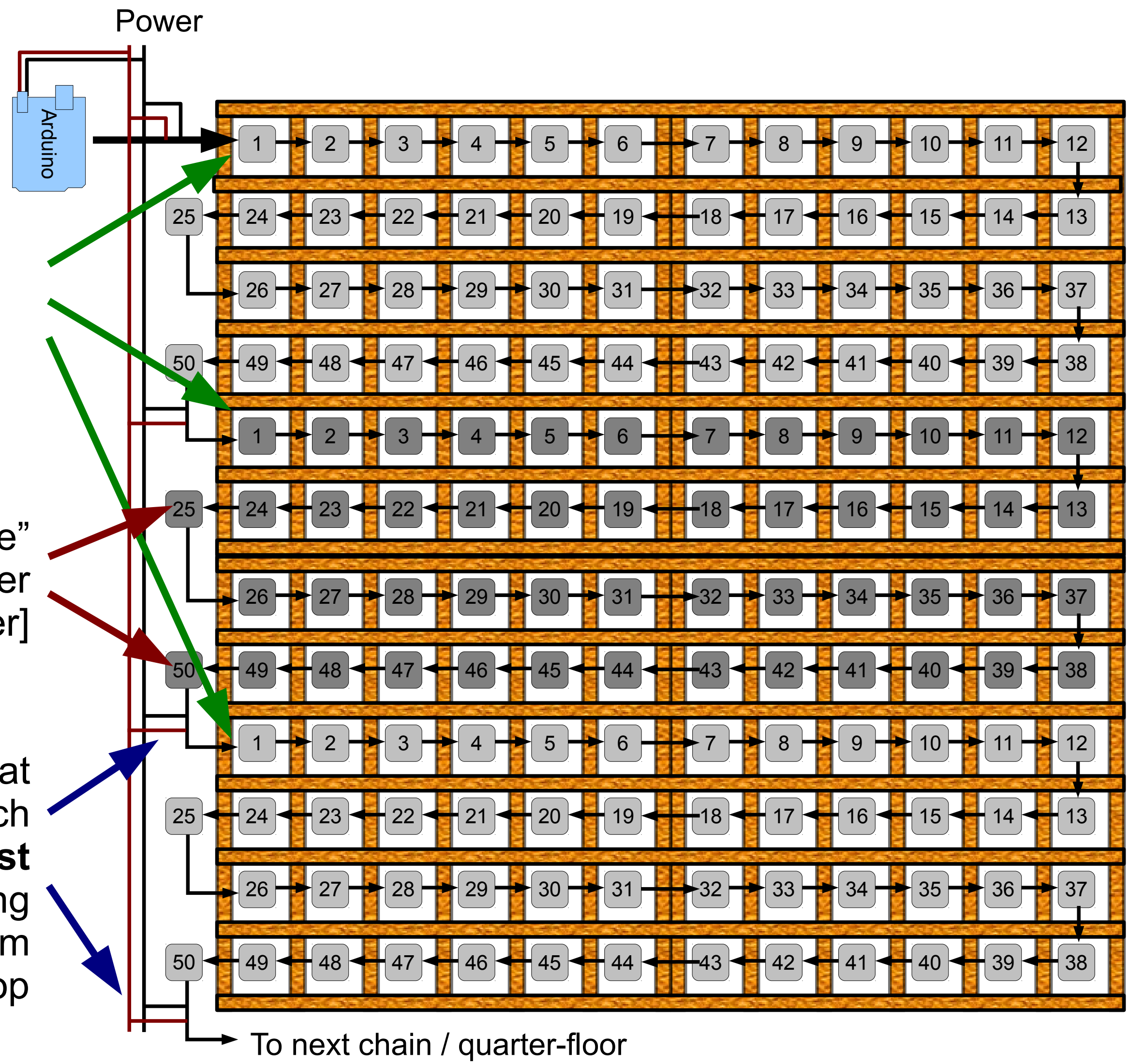




Main floor assembly – WS2801 LEDs

Our LEDs from pcboard.ca come in chains of 50. We should be able to just lay them into the floor, connect them all into one long chain from the Arduino, and add some extra power

Data+
Clock

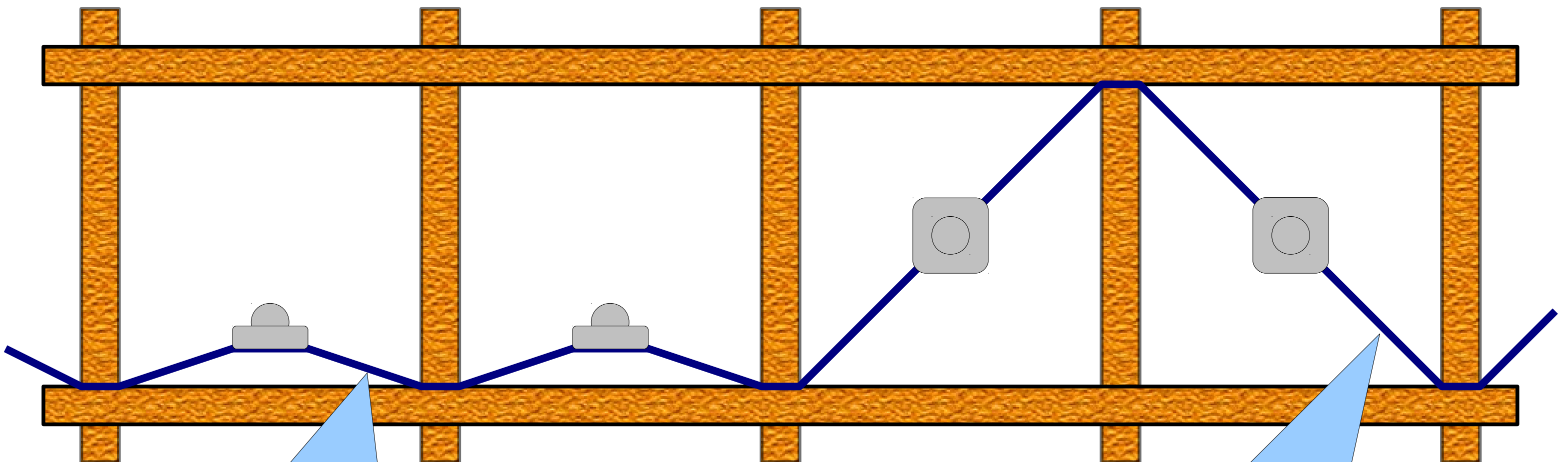


Each quarter-floor uses 3 chains of 50 LEDs (12x50 total)

Note there are 2 “spare” LEDs per chain, 6 per quarter, 24 total. [See later]

We also “top up” the power at the beginning / end of each chain, **including the last one**, otherwise really long chains suffer “brownout” from gradual voltage drop

WS2801 cabling detail



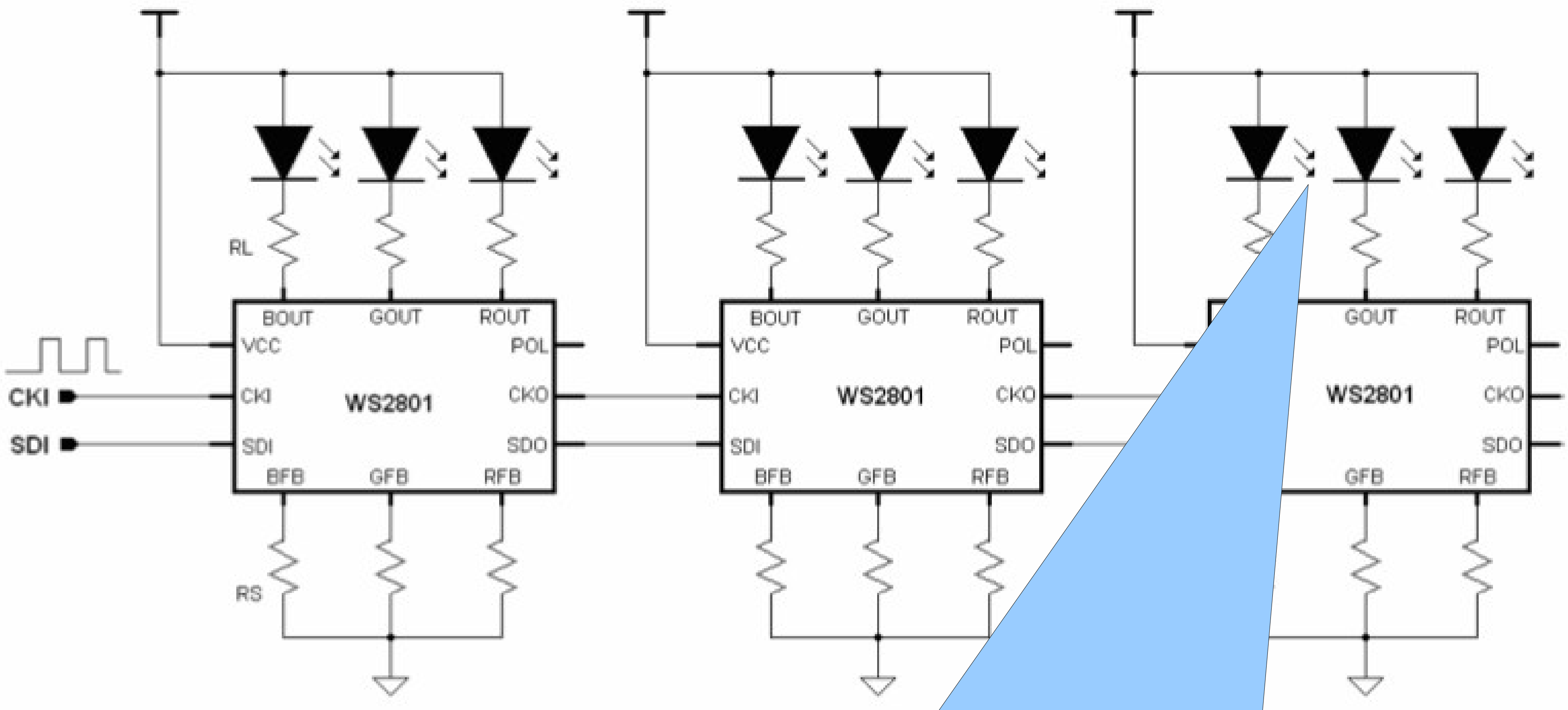
If our LEDs were just over 4in centre-to-centre, like our robotshop ones, we would lay the LEDs down one side, and use the slack to aim them nicely. This actually works OK.

... but with our pcboard.ca chains we actually have tonnes of slack to play with, And can weave diagonally, hanging the LEDs much closer to the centre, and there's STILL loads of slack!

Our pcboard.ca LEDs are also NOT plastic-coated. This is no concern as we will be hiding them under all that wood and acrylic, it saves us a little $$, but also...

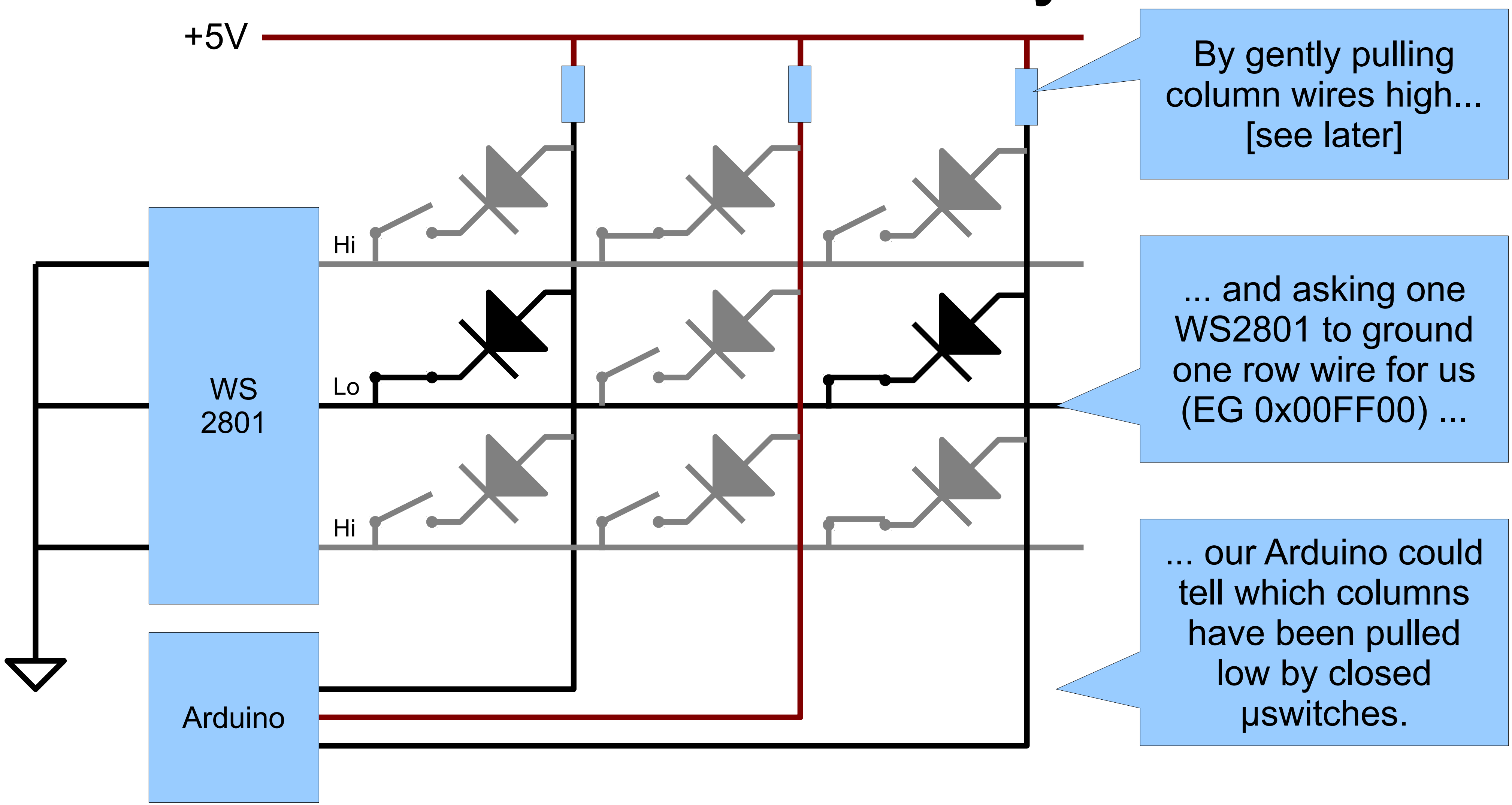
Remember those “spare” LEDs?

We could (ab)use the un-coated WS2801 chips / boards for other purposes!

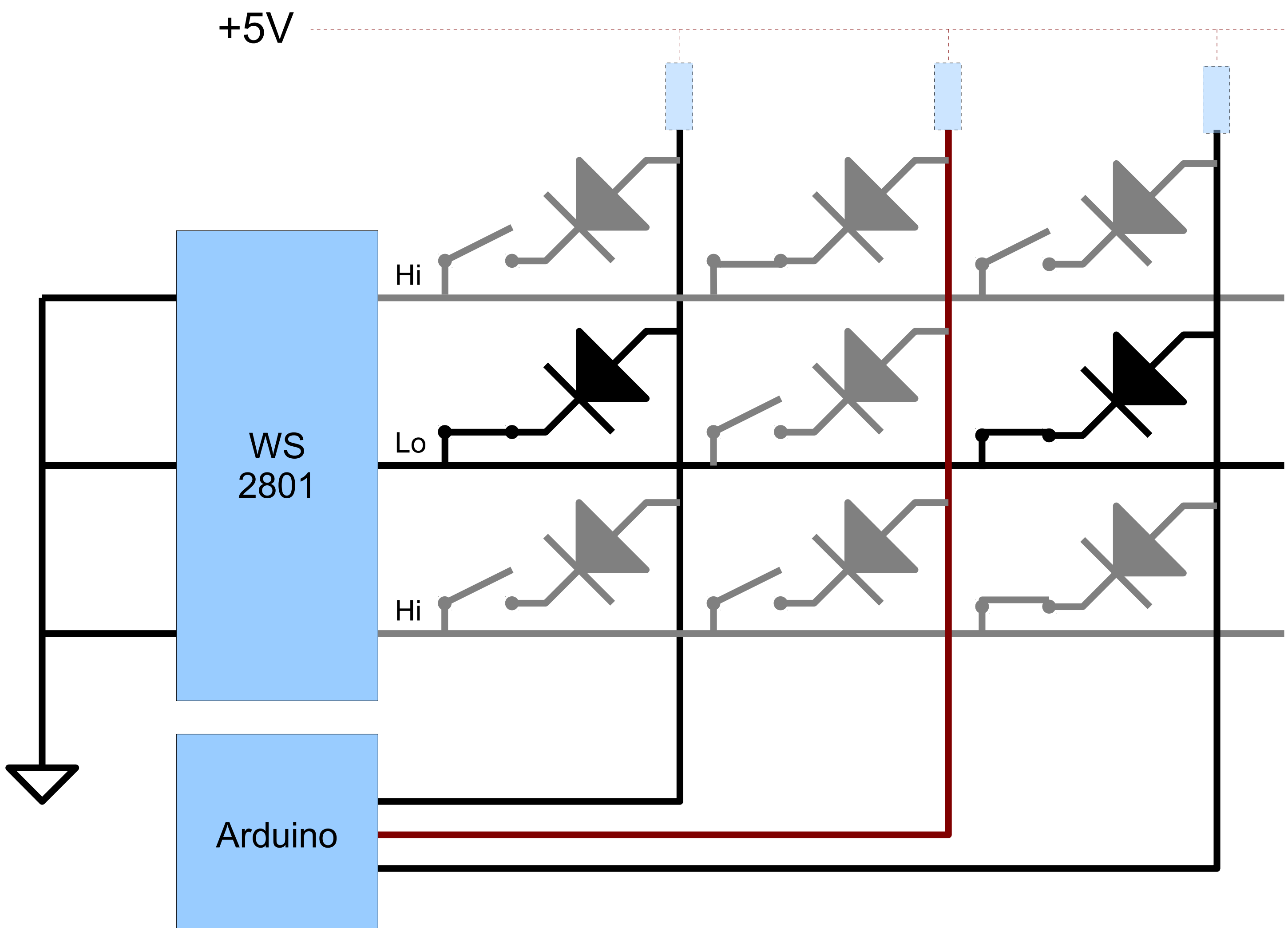


There's no reason why these HAVE to be LEDs, the chips could drain any load, subject to maximum current limitations

Sensor circuitry

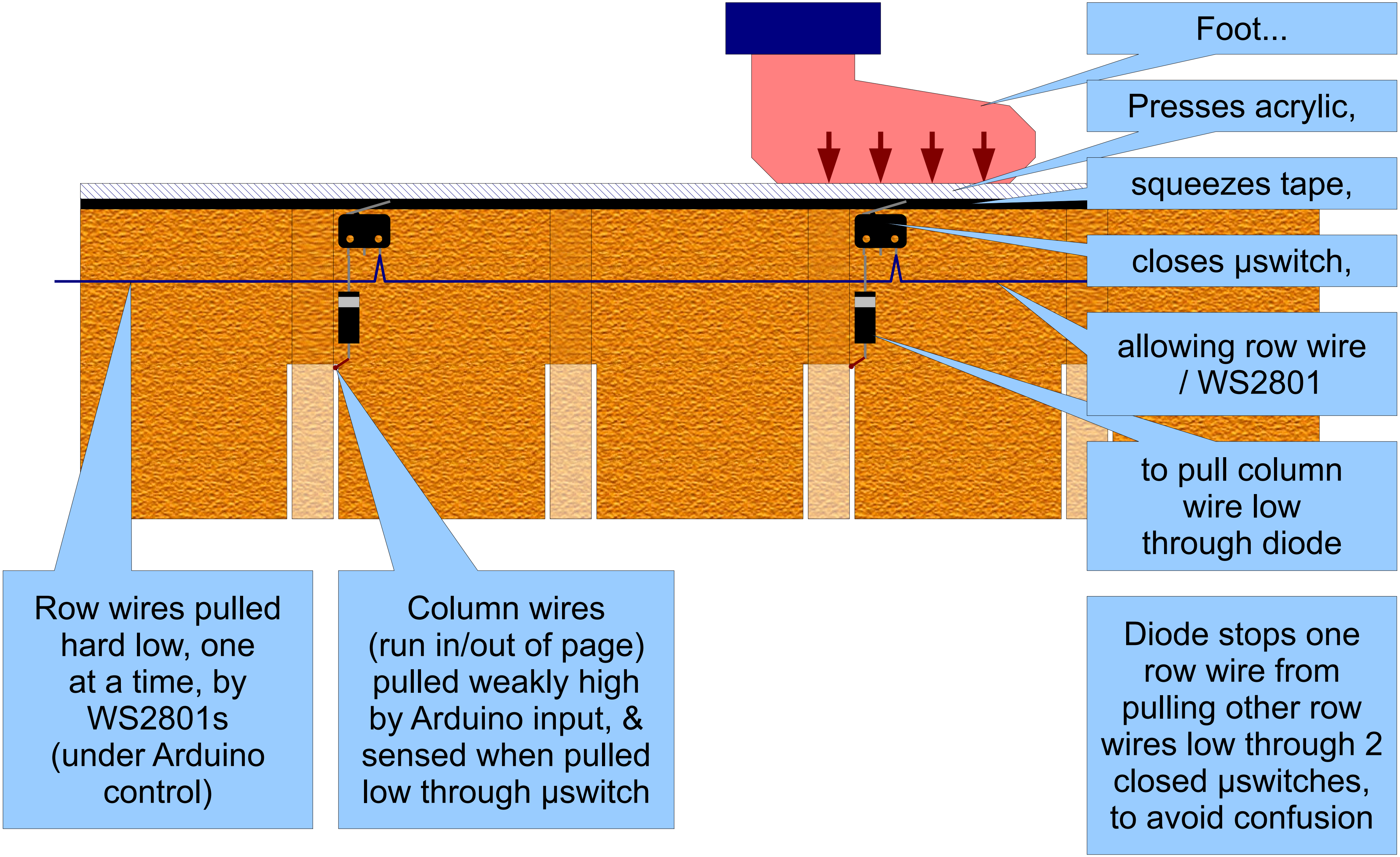


Sensor circuitry



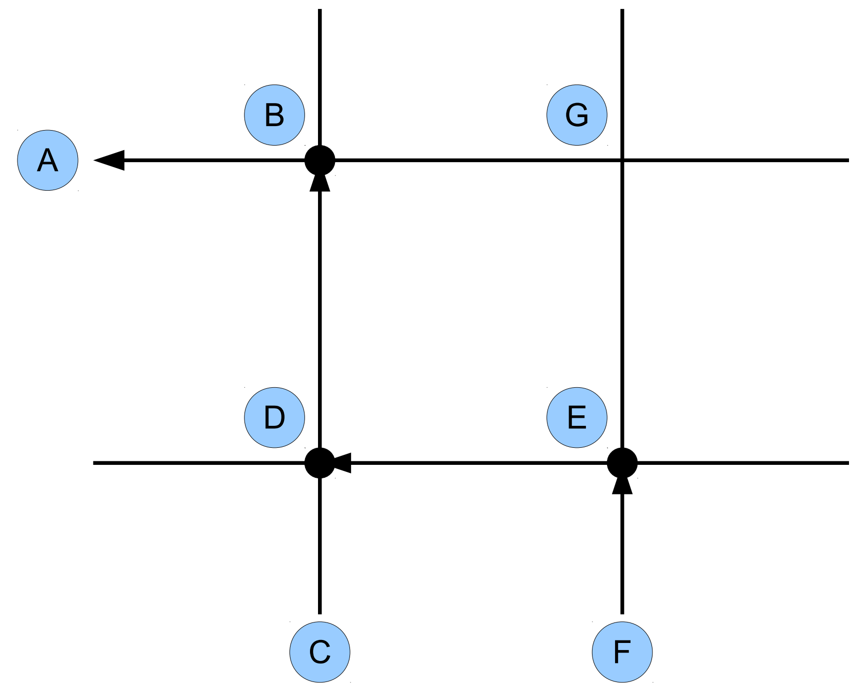
The Arduino can actually pull input lines high for you too,
so we don't even need these pull-up resistors, just
row/column wires, microswitches, diodes, and WS2801 abuse

Microswitch Sensors Detail



Without diodes...

- If row A is being pulled low to sense those µswitches...
- and B is closed, it correctly pulls column C low, however...
- If D and E are also closed, column F is also pulled low through the indicated path
- Incorrectly implying µswitch G is also closed
- The diodes block this behaviour at D
- A **row** can then pull **columns** low, but not vice-versa.



Isn't that too much hassle?

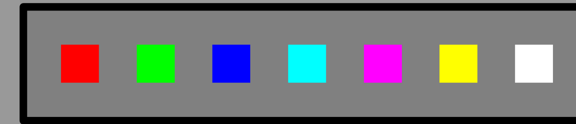
This may remind you of the row/column wiring for v1 / v2 which was too impractical? Important differences:

- This is 12x12, not 24x24
- 1 wire per 2 rows, 1 wire per 2 columns, not 1 wire per row and 3 per column
- 3 **nicely-spaced** solder joints per switch (12x12x3 = 432, not 24x24x4 = 2304, <20%)
- No fabrication of custom LED-holders
- Can do some (most?) of the soldering outside the floor, not in-situ and at awkward angles.
- Much bigger spacing, much less chance of shorts, much easier to trace shorts, could probably even write Arduino code to detect where!

Some game ideas

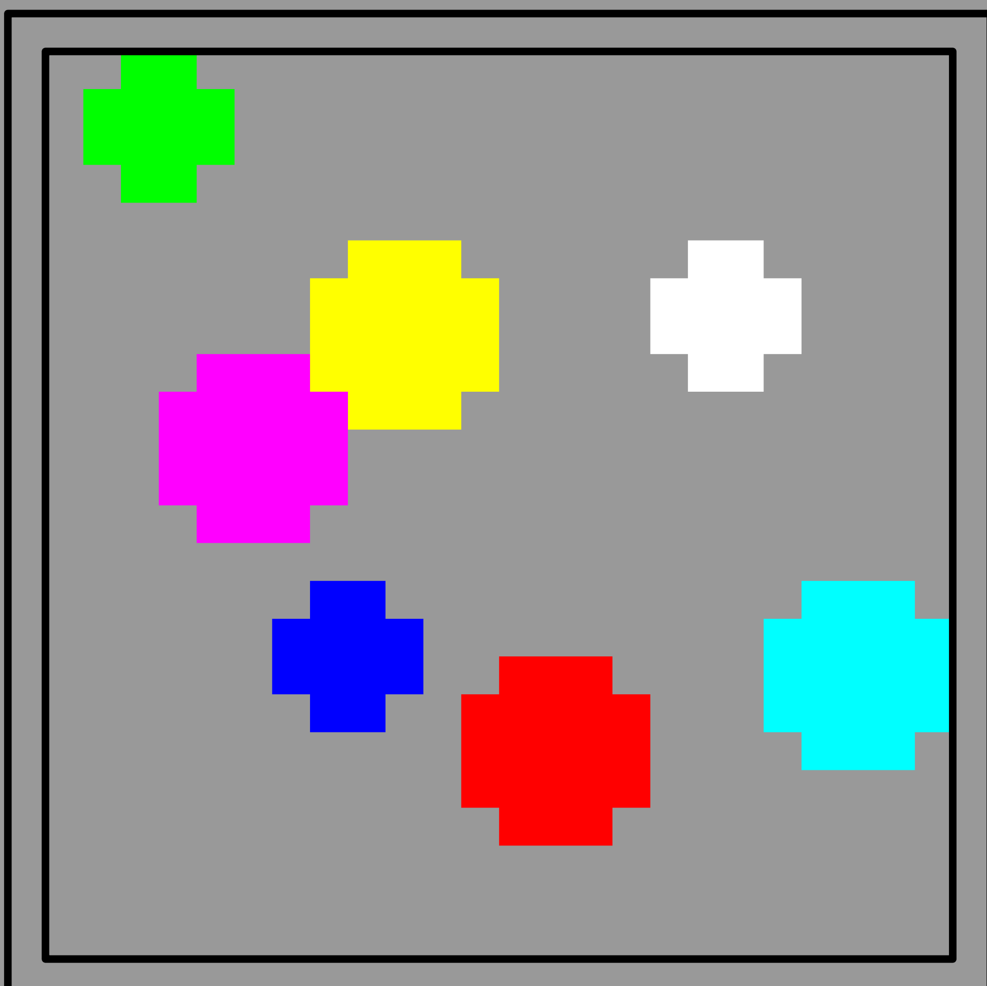
(See also http://noseynick.net/leddf/designer/)

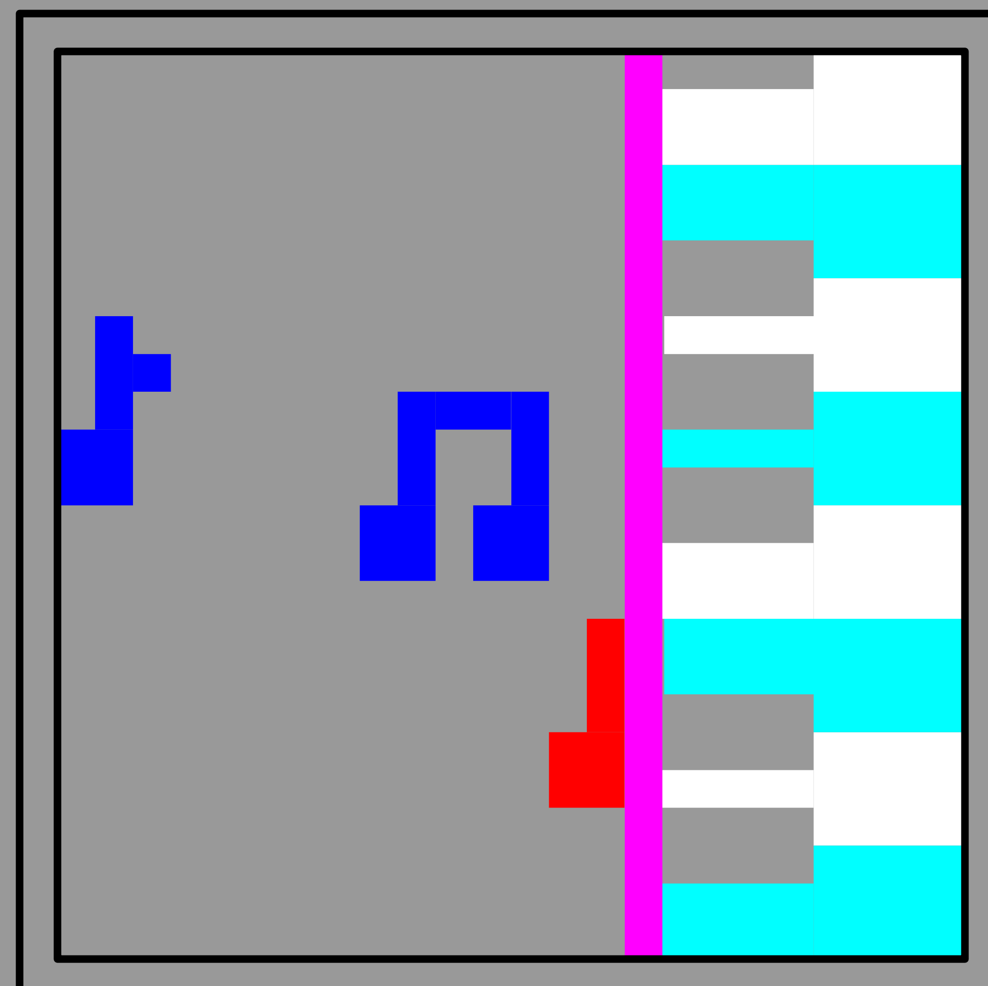
Some game ideas



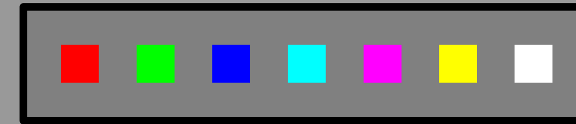
Dot Chase (by Aiden): Dots move around. Chase the dots of your colour, jump on them, they will reappear elsewhere. Player who stomped their own colour in the time limit the most wins. Maybe size handicap for good players?

Music time! (Nick's idea) – jump on the indicated piano keys in time with the music – a bit like “Rock Band” or the newer “Piano Master”



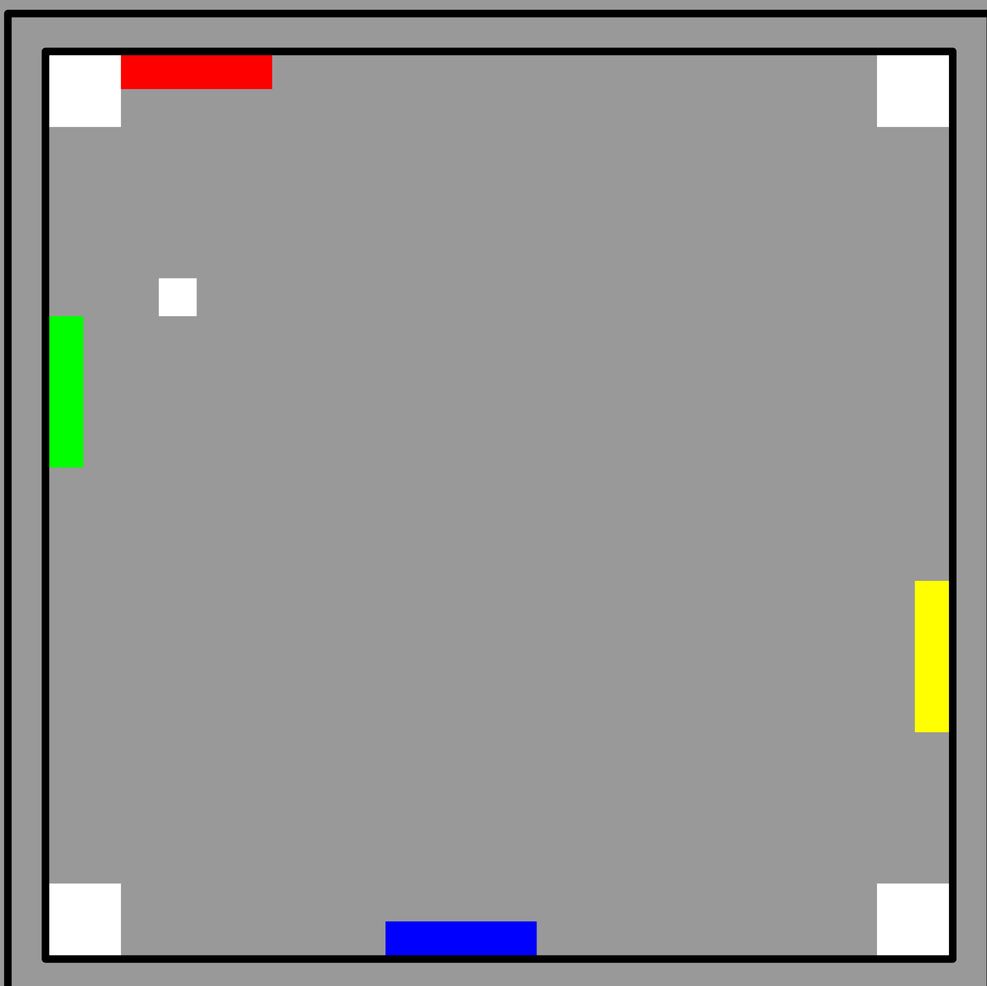


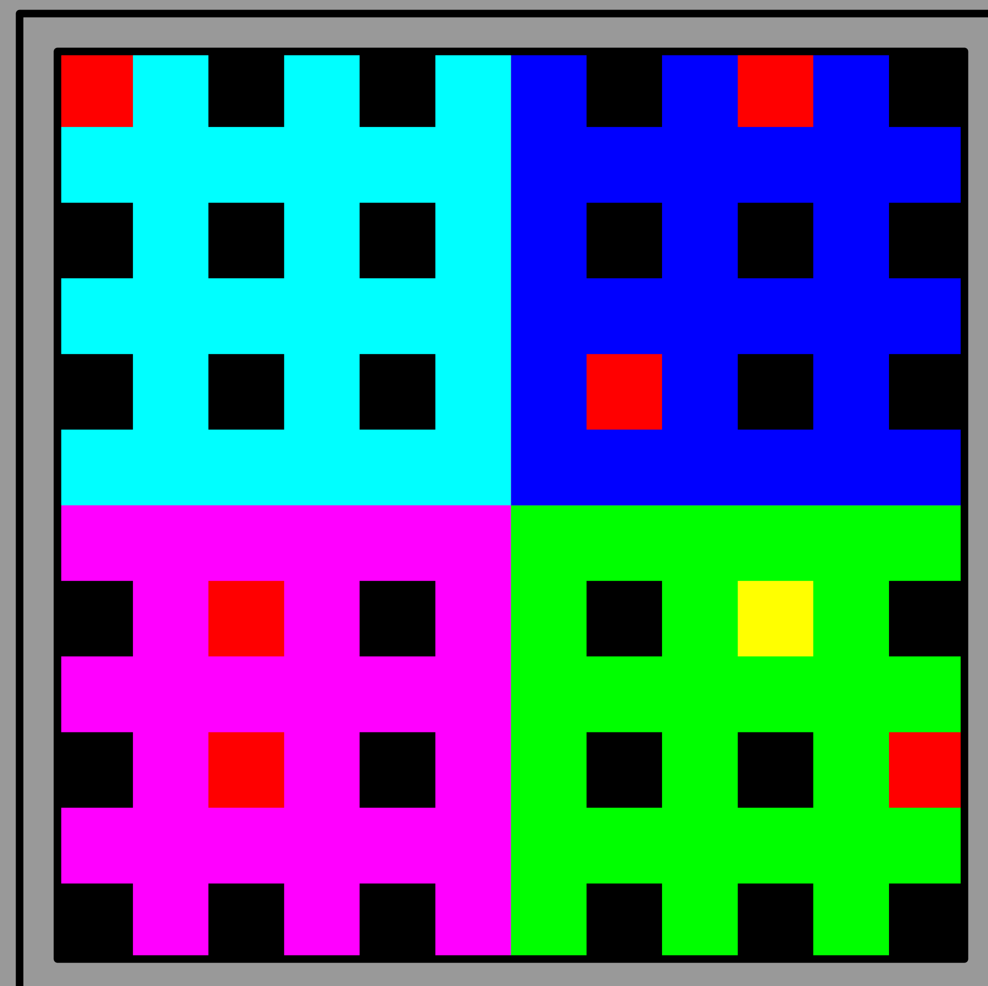
Some game ideas



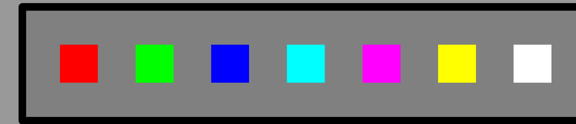
Multi-Player-Pong! Obvious one really

Fire-brigade, by KJ. Windows in the 4 buildings light up with red/yellow flames. Step on them to put them out

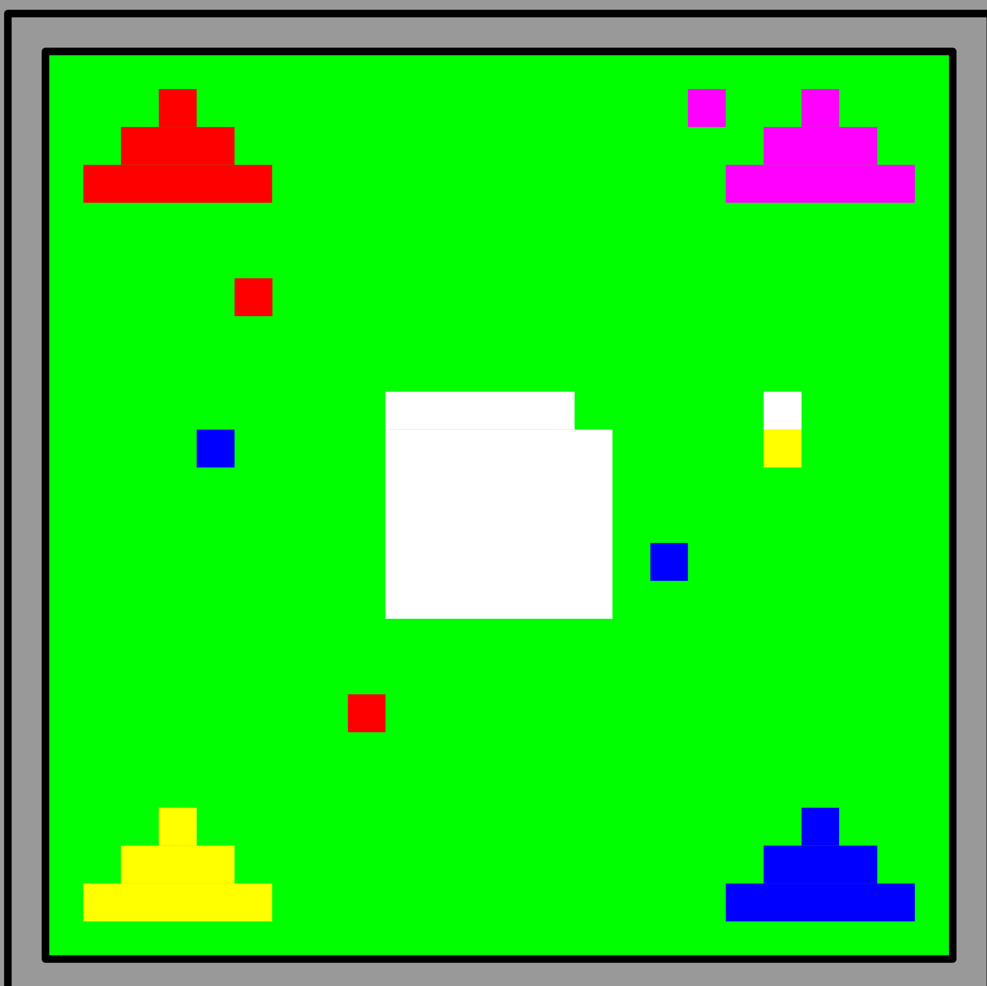




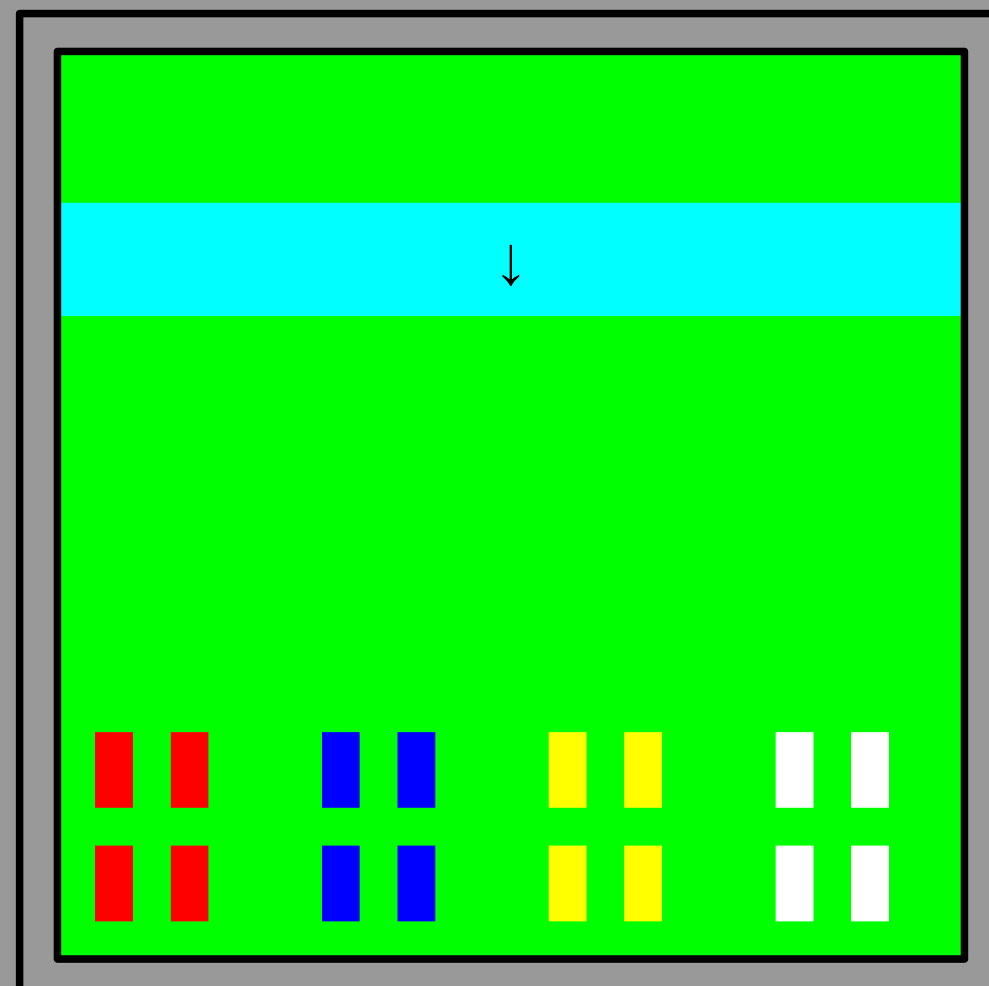
Some game ideas



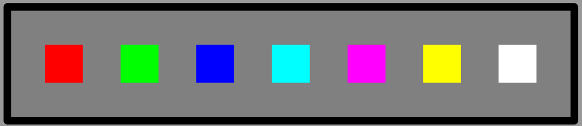
Ant Attack (by KJ) – ants (possibly single-pixel?) appear out of the hills and crawl around in a spiral to the food on the picnic blanket in the middle (taking some white with them?), and back to their nests. Squish the ants!



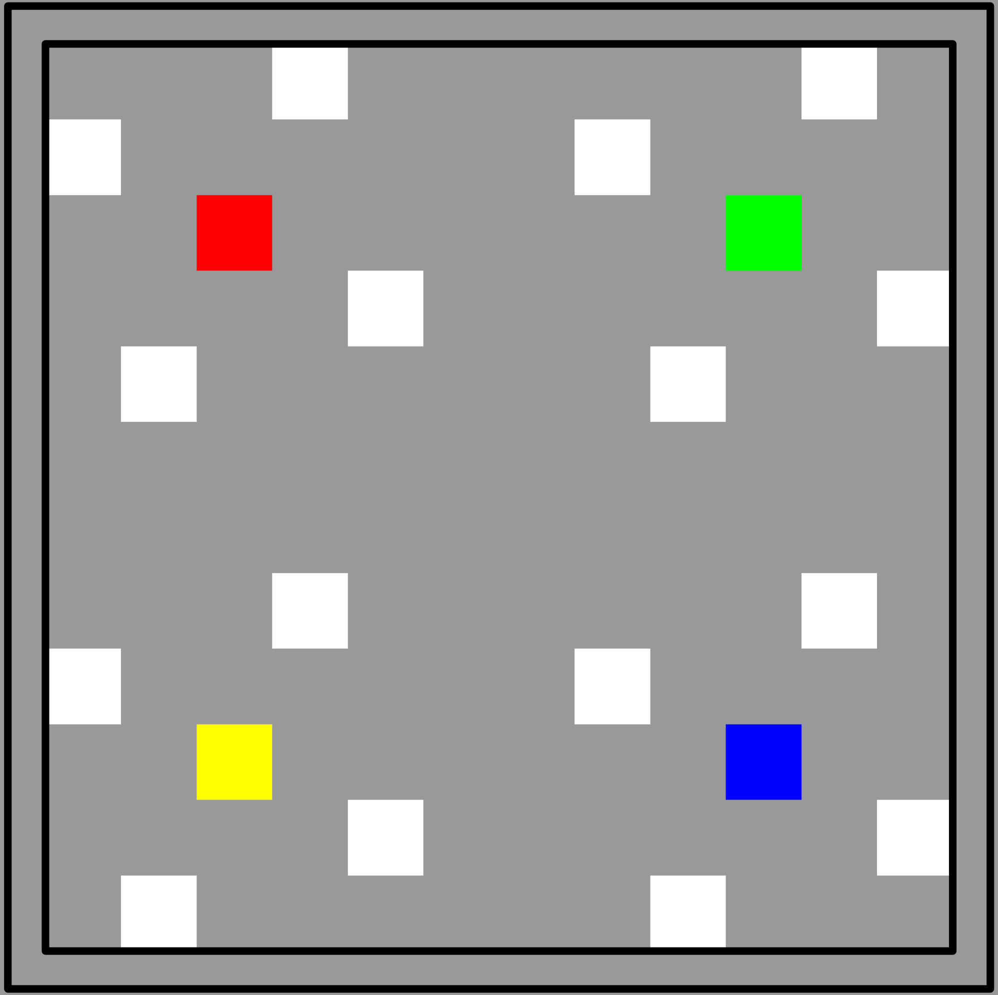
Horse jump (by Estelle, for KJ): Players kneel on all fours at one end of the floor. Obstacles come down the screen and you must jump over them



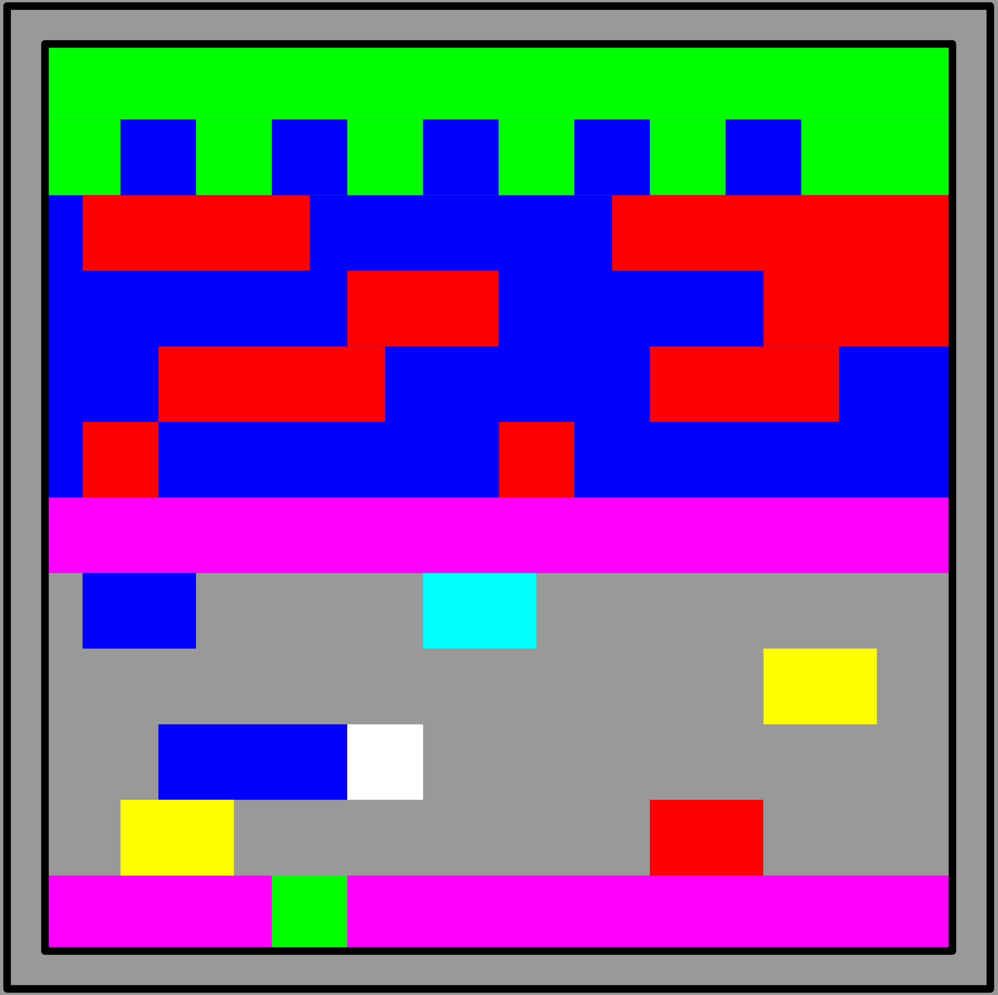
Some game ideas



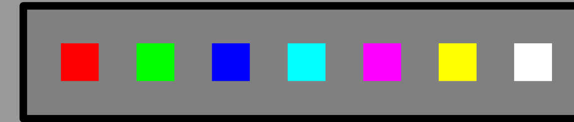
Dance-off – one player dances, their moves scroll across the board to another player who has to copy them as accurately as possible. Repeat a few times for different players sending to different players. 1-player or 2-pl cooperative DDR-like also possible



Frogger? These frogs and cars made out of small numbers of pixels are tricky! We hope they will look a bit more obvious once they are moving.

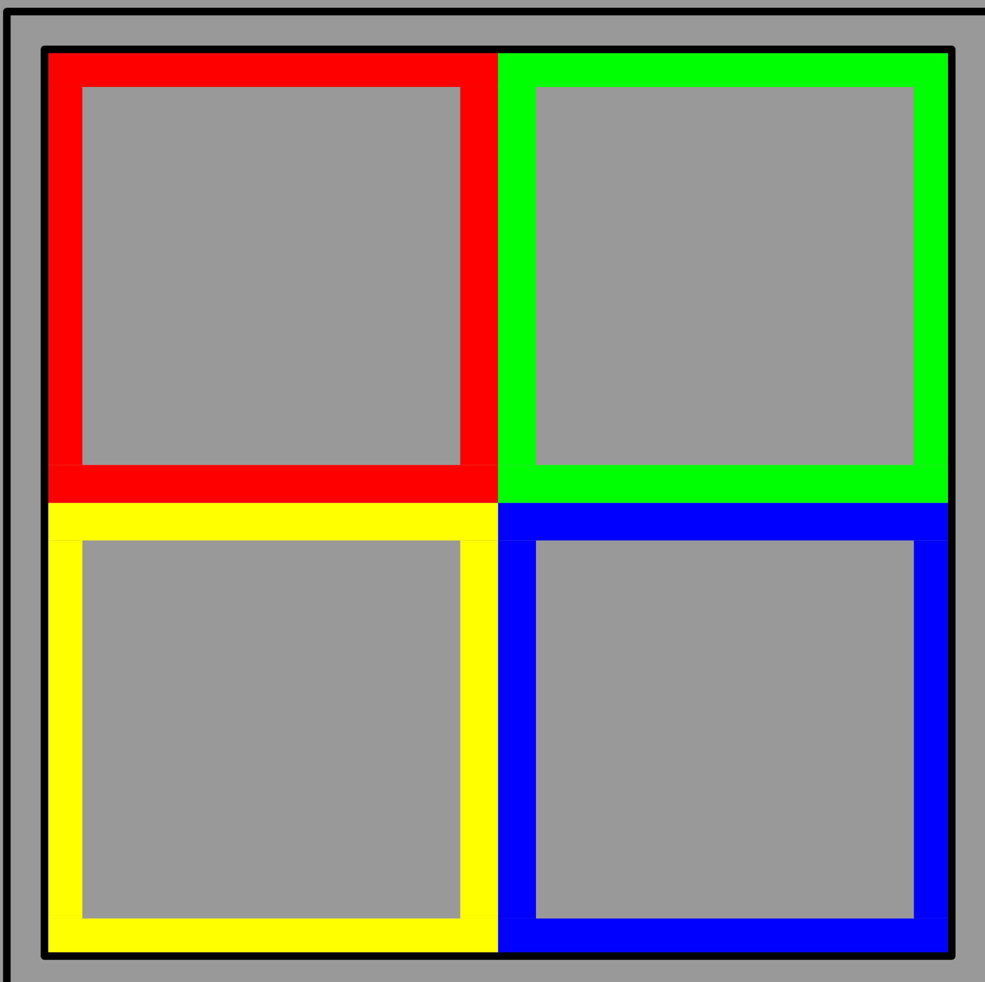


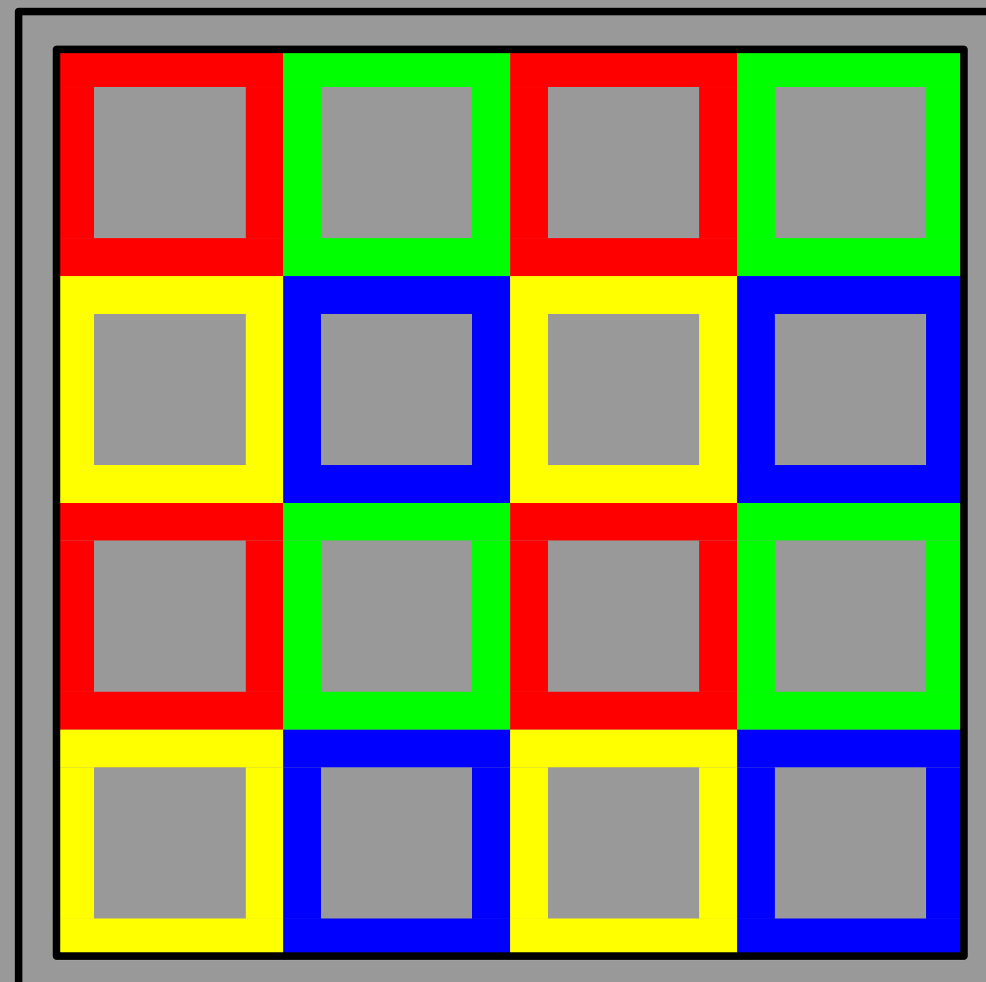
Some game ideas



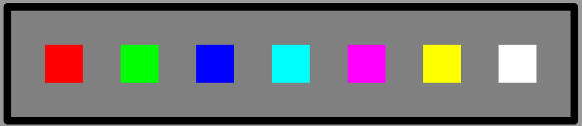
Simon... Seriously, you need me to explain Simon?

Multi-player Simon – same patterns, see who remembers the longest sequence?



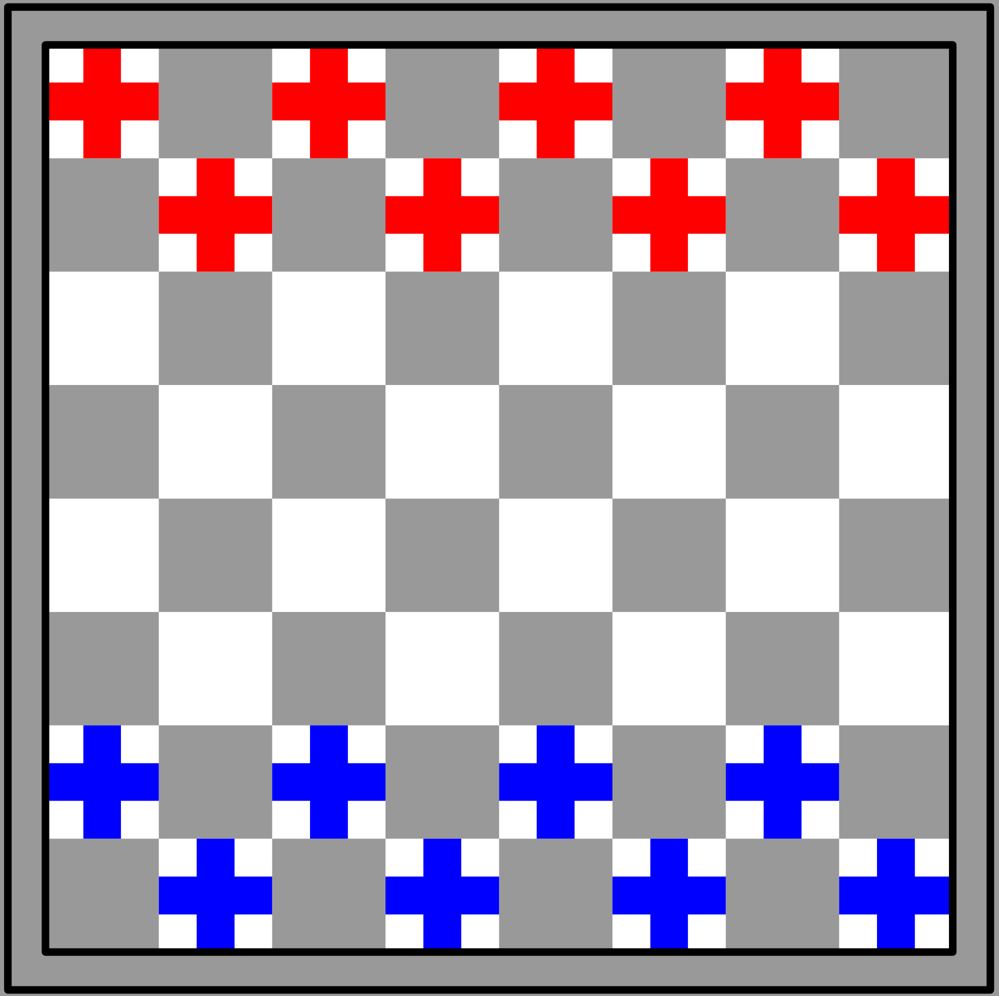


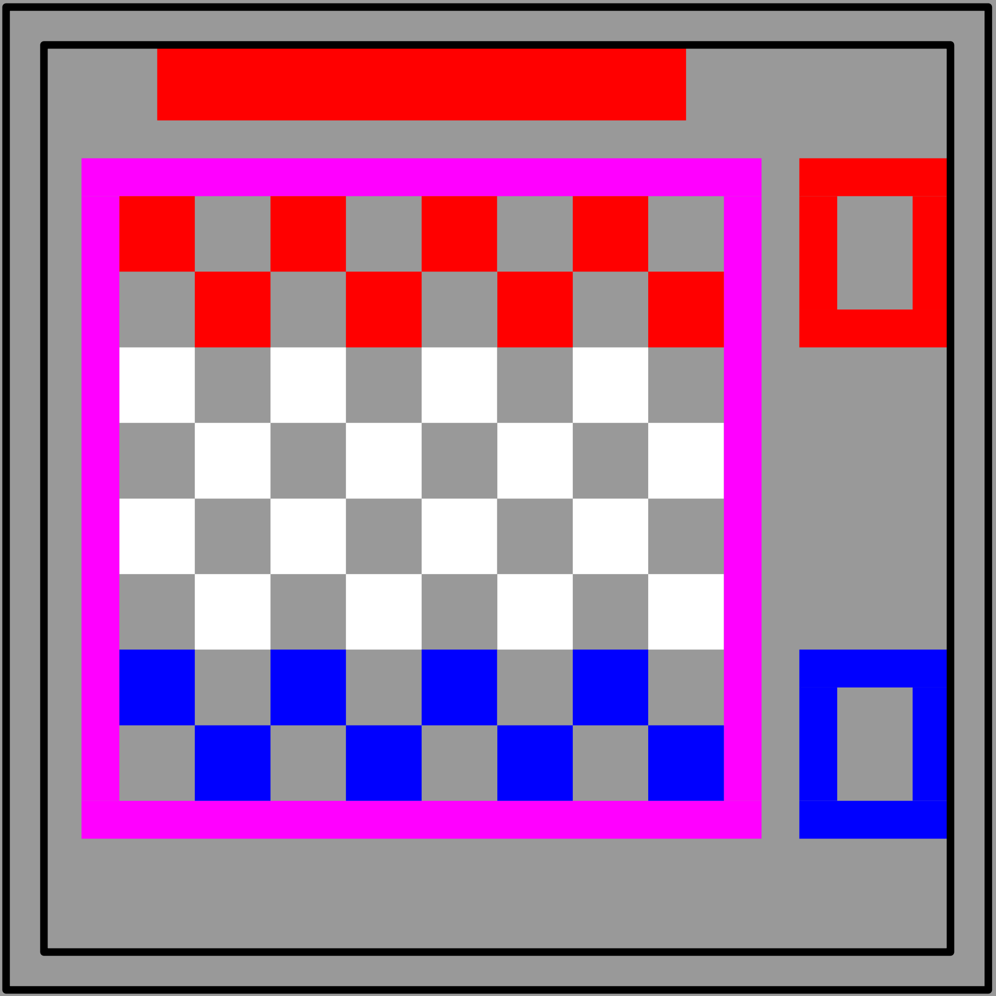
Some game ideas



For a more relaxing low-exercise game...
Checkers / Draughts, anyone? Problem here is the squares are 3x3 and the sensors are 2x2.
Chess pieces in 3x3 would be REALLY tricky, but KJ's going to give it a try

Instead, use some border space for scoring or something? Squares are now the same size as the sensors, but obviously no detail for chess pieces this small, only Checkers / Draughts





More game ideas

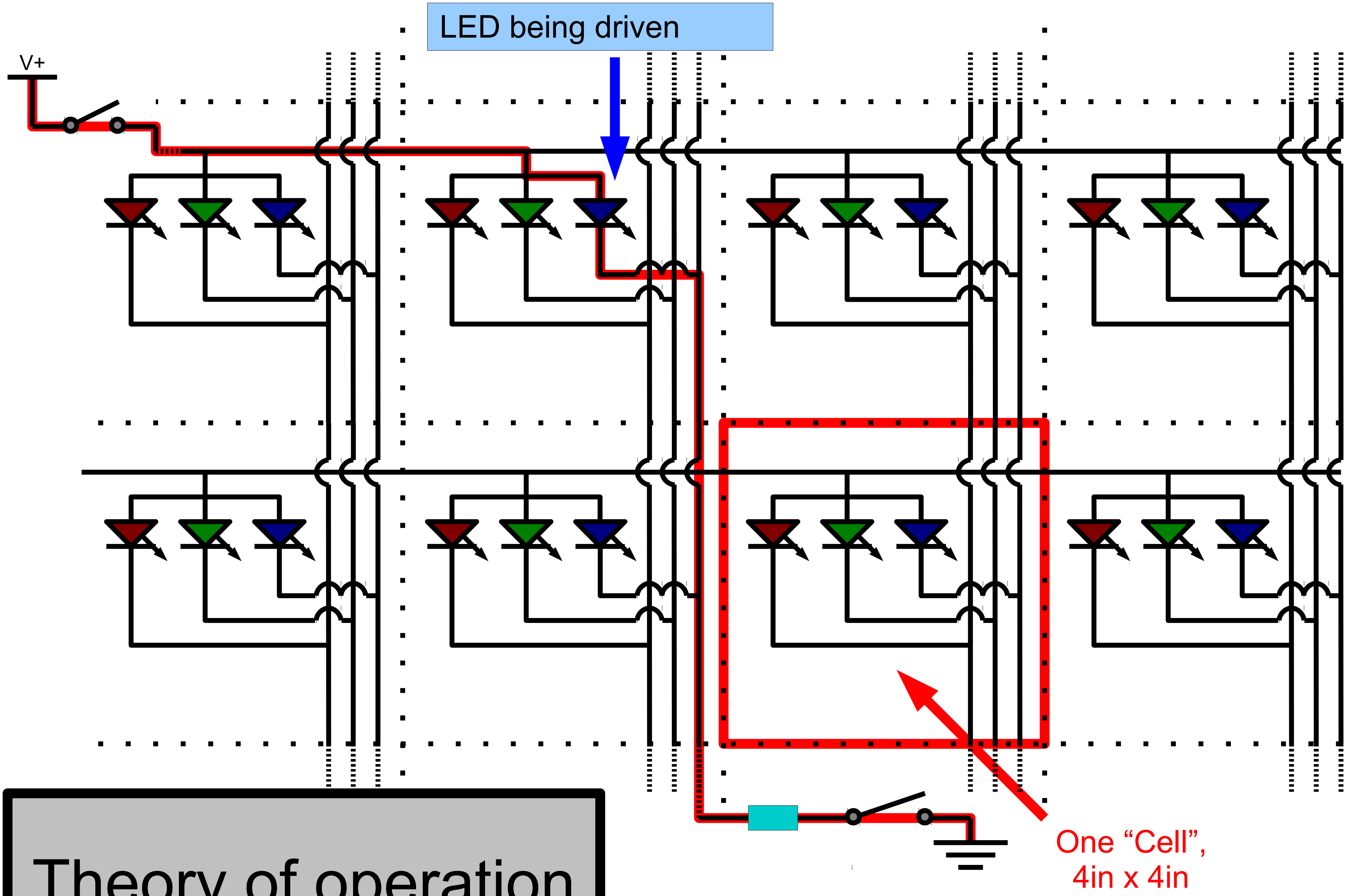
http://noseynick.net/leddf/designer/

Older slides,
Historical / test stuff,
feel free to ignore!

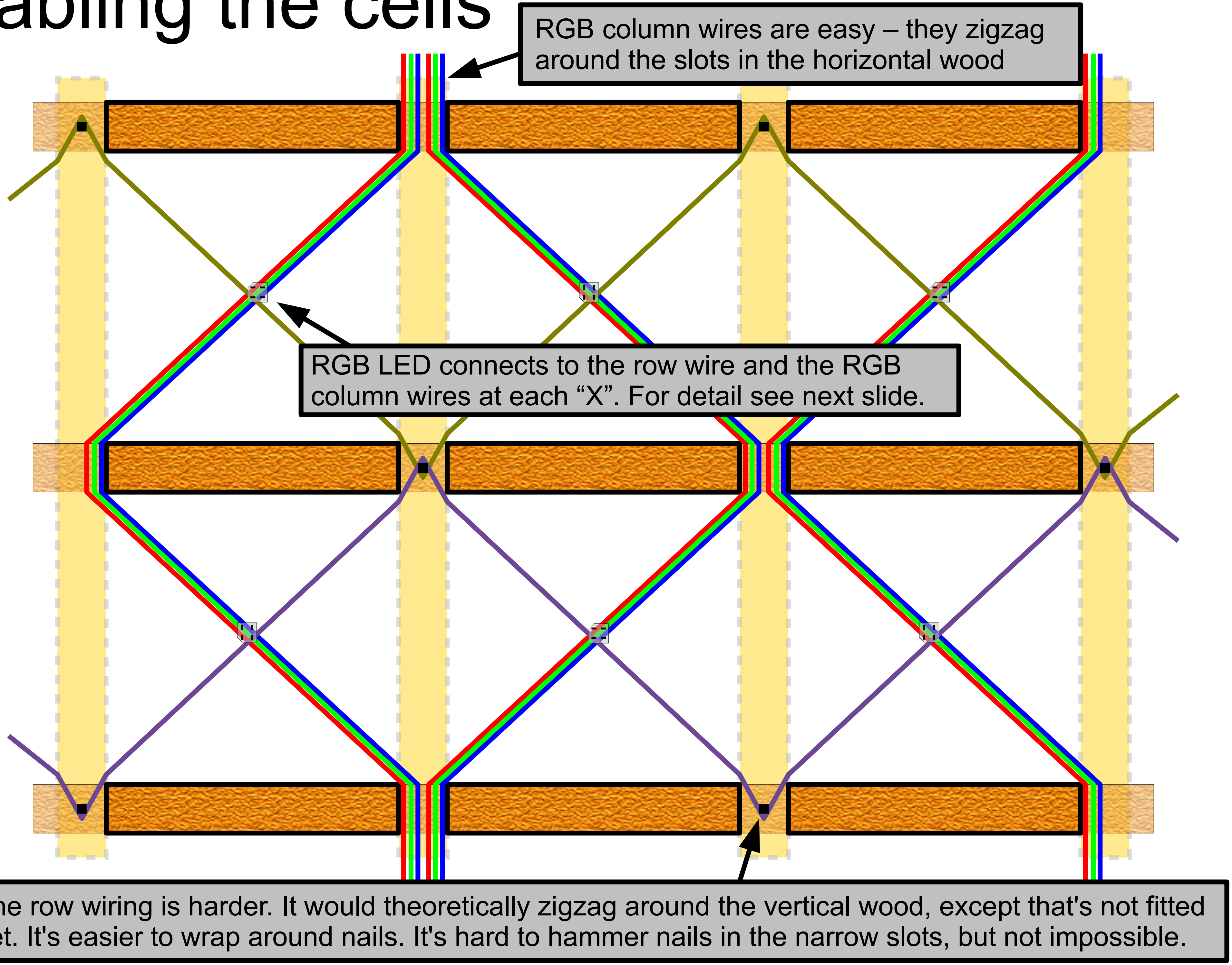
LED Dance Floor v1 Wiring

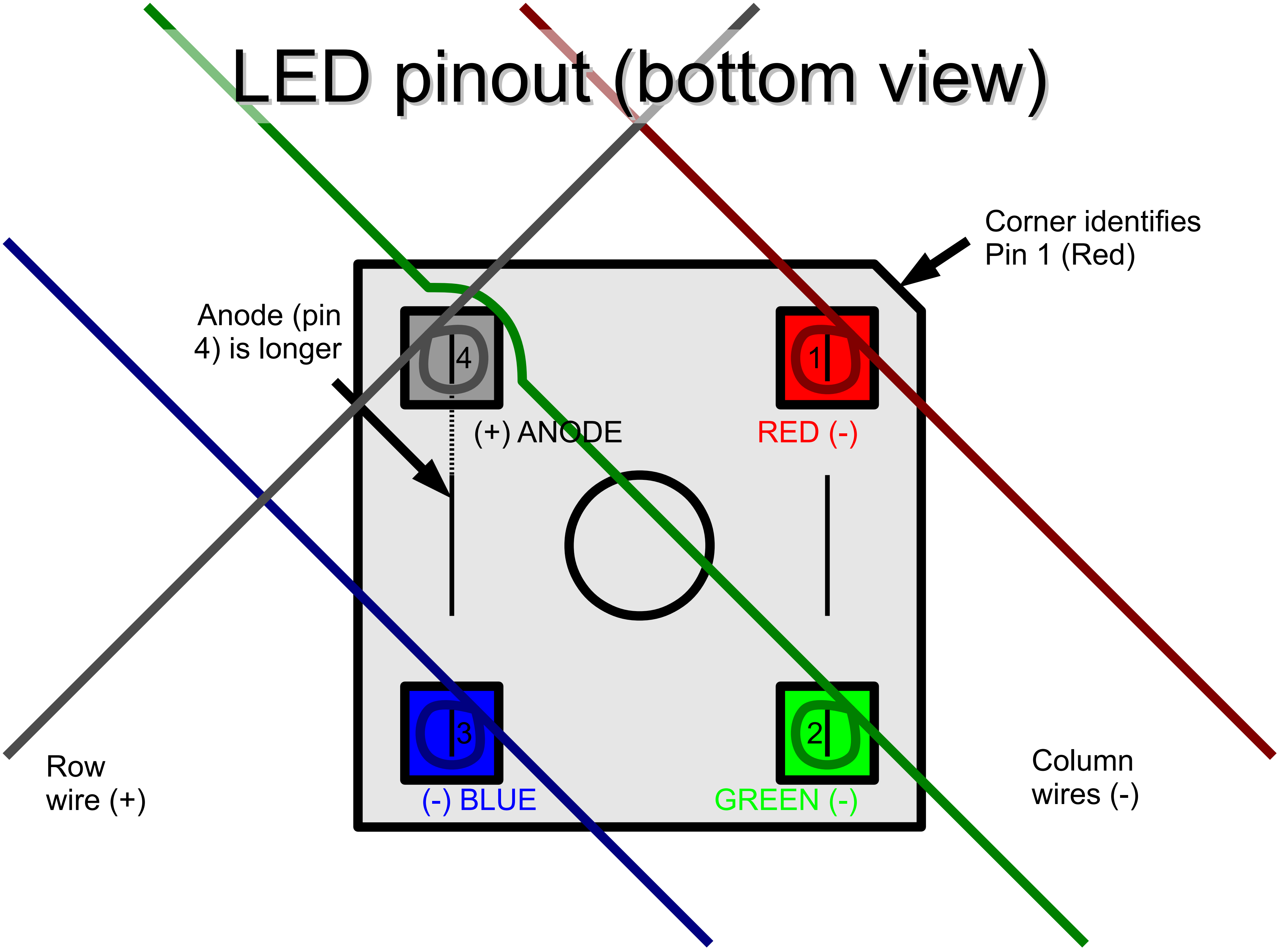
- Our first Dance Floor design was going to use Piranha ultra-bright RGB LEDs.
- We ran 3 wires down each column (R,G,B), one wire across each row (common anode)
- We had a complicated collection of shift-registers, transistors, and resistors to power 1 row whilst grounding the corresponding RGB columns to display the row contents.
- Whilst displaying 1 row, we were clocking the next row's contents into the shift registers
- If we display each row in quick succession, persistence-of-vision makes us see the entire display (but at a corresponding loss of brightness!)

Theory of operation



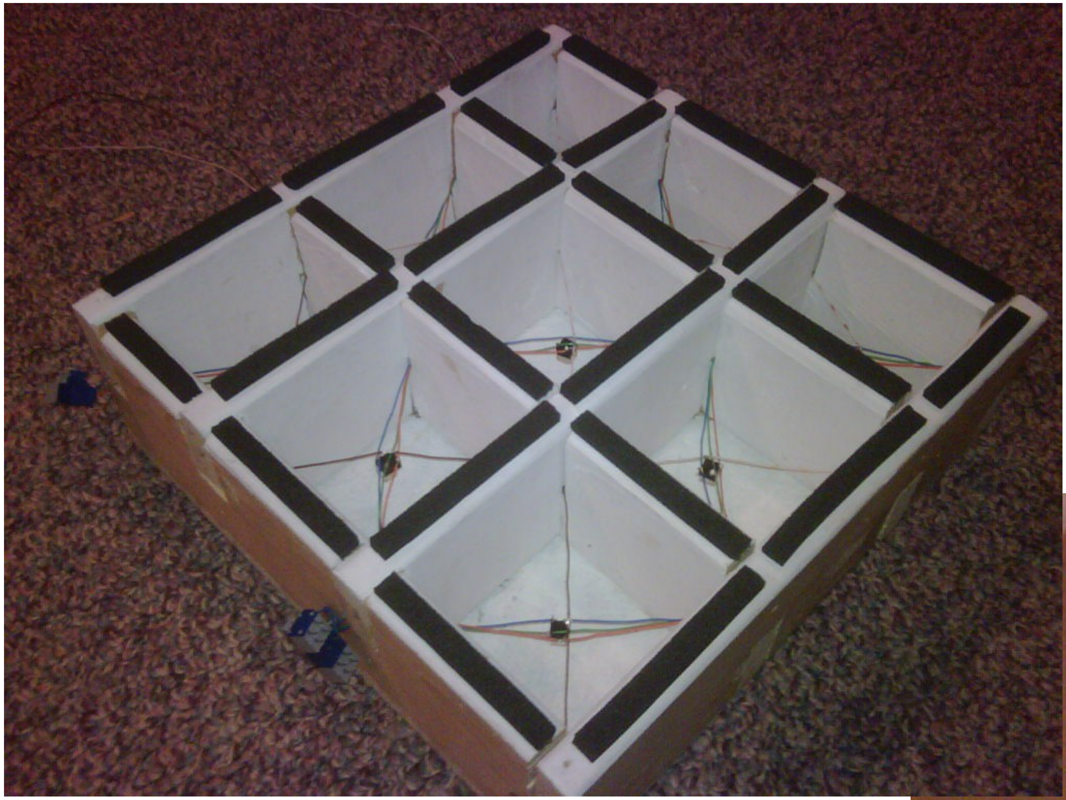
Cabling the cells

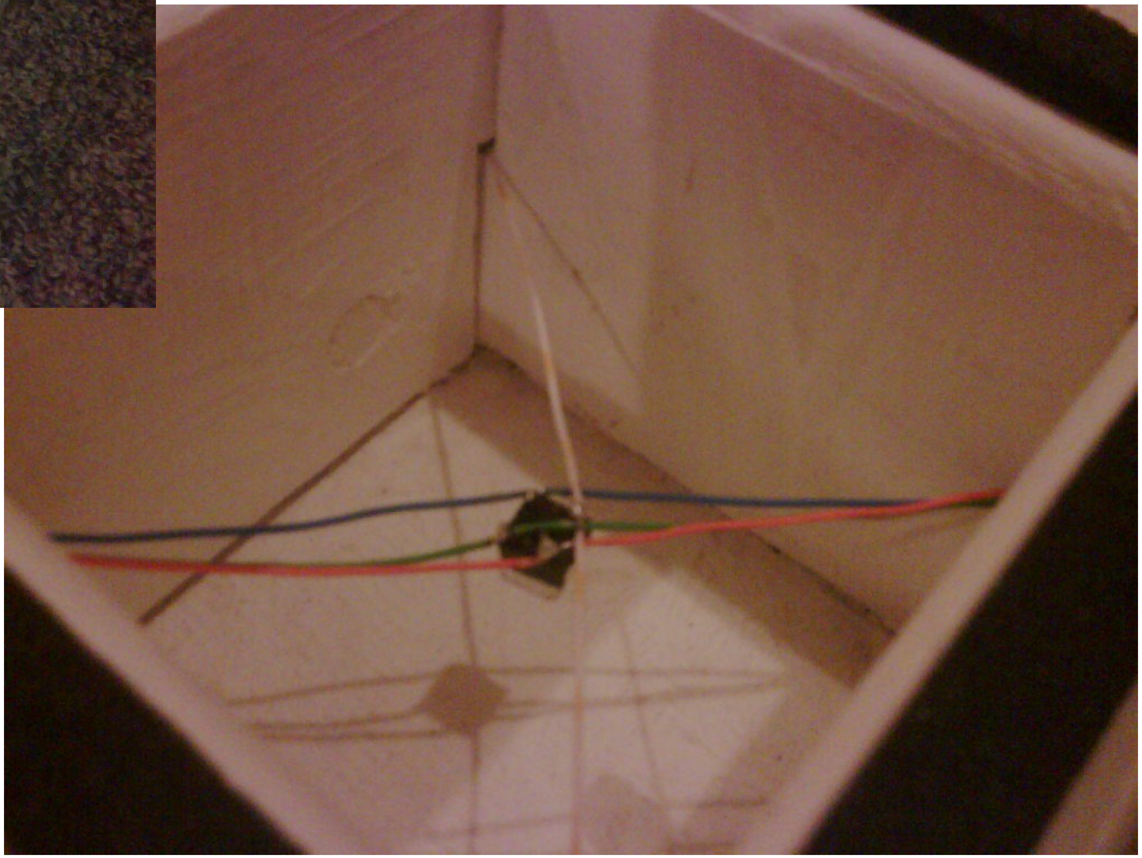




LED pinout (bottom view)
Corner identifies
Pin 1 (Red)
Anode (pin
4) is longer
4
1
(+) ANODE
RED (-)
3
2
Row
wire (+)
Column
wires (-)
(-) BLUE
GREEN (-)

Cabling the cells





Control via 74HC595 shift/latch registers

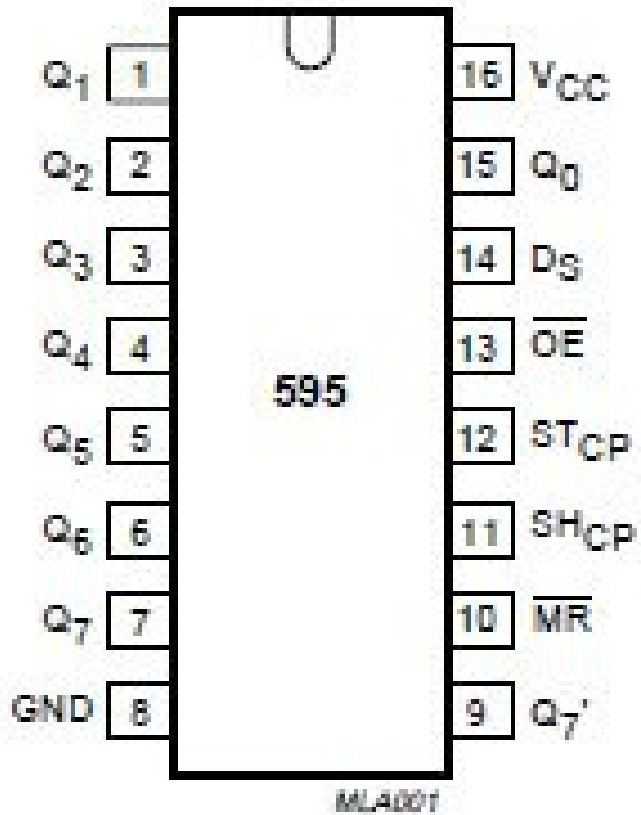


Fig.1 Pin configuration.

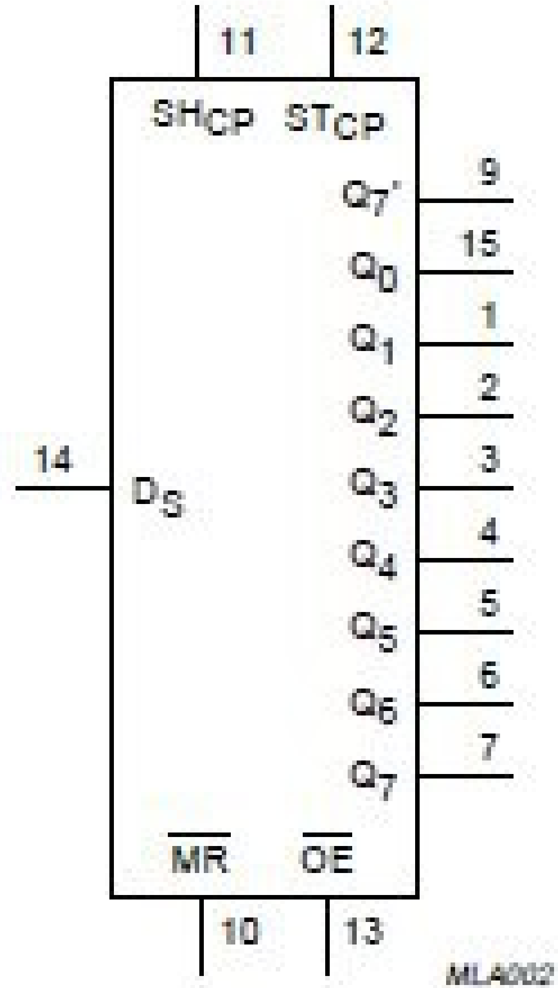
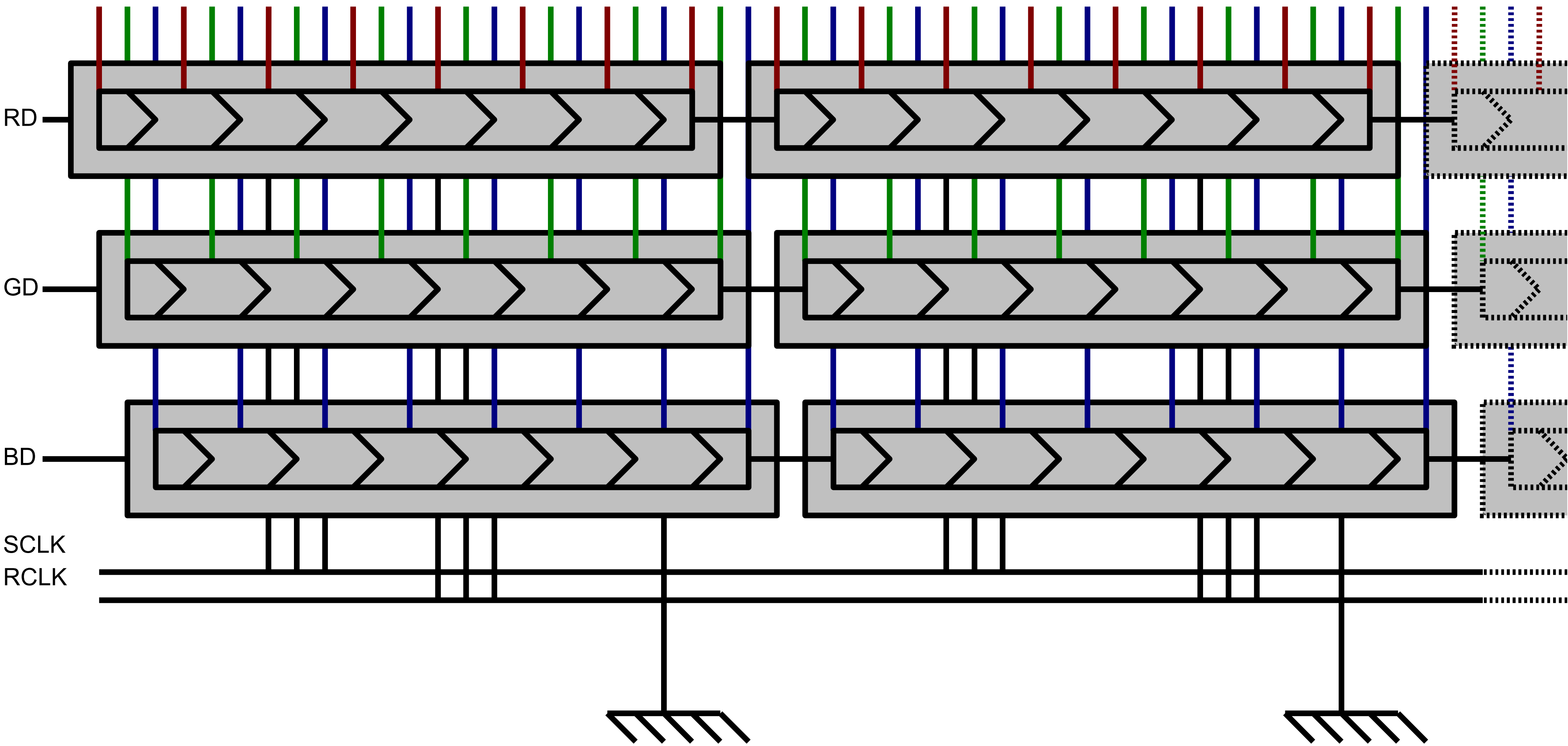


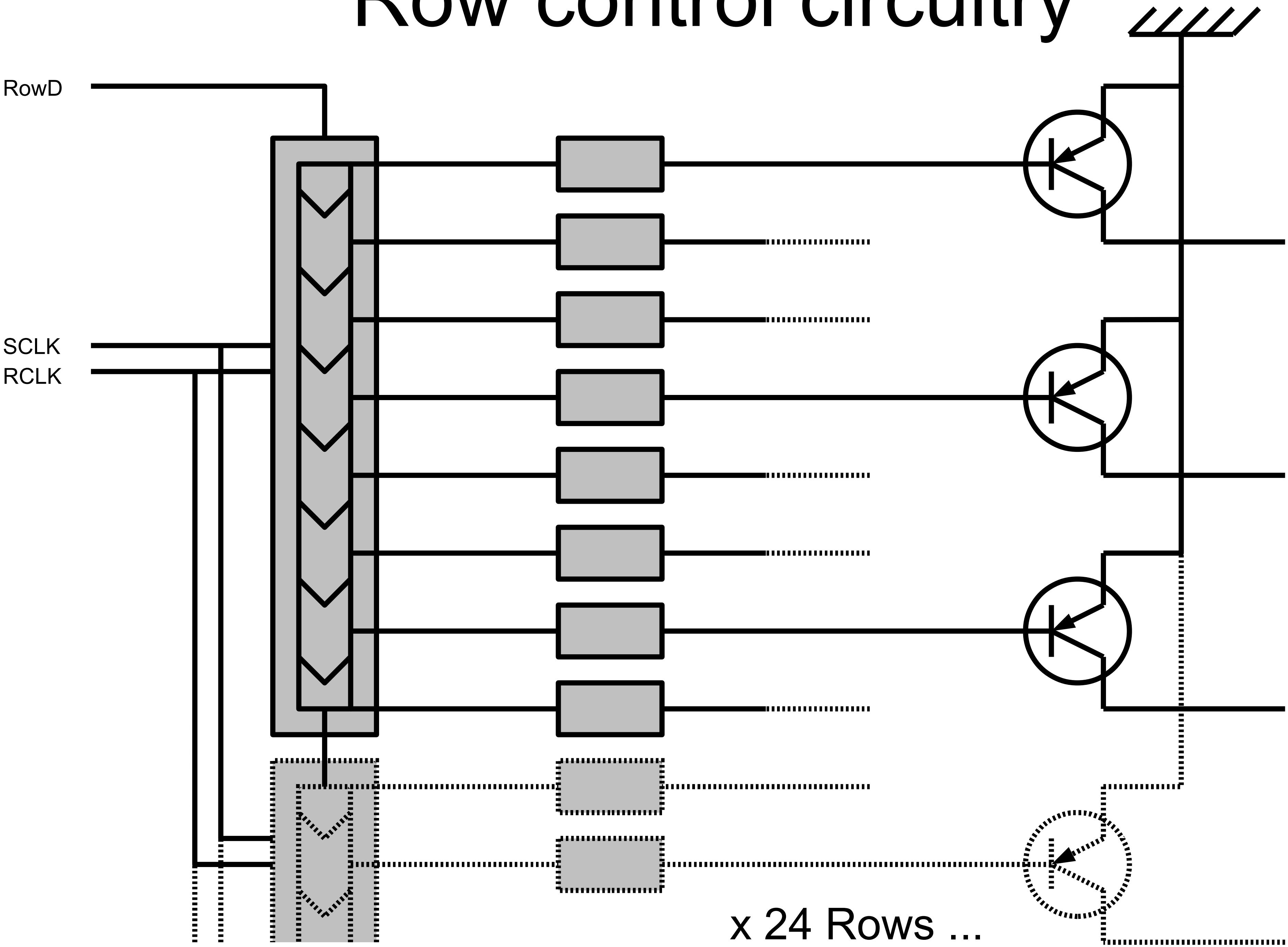
Fig.2 Logic symbol.

Column control circuitry

24 Cols x (R+G+B) ...



Row control circuitry



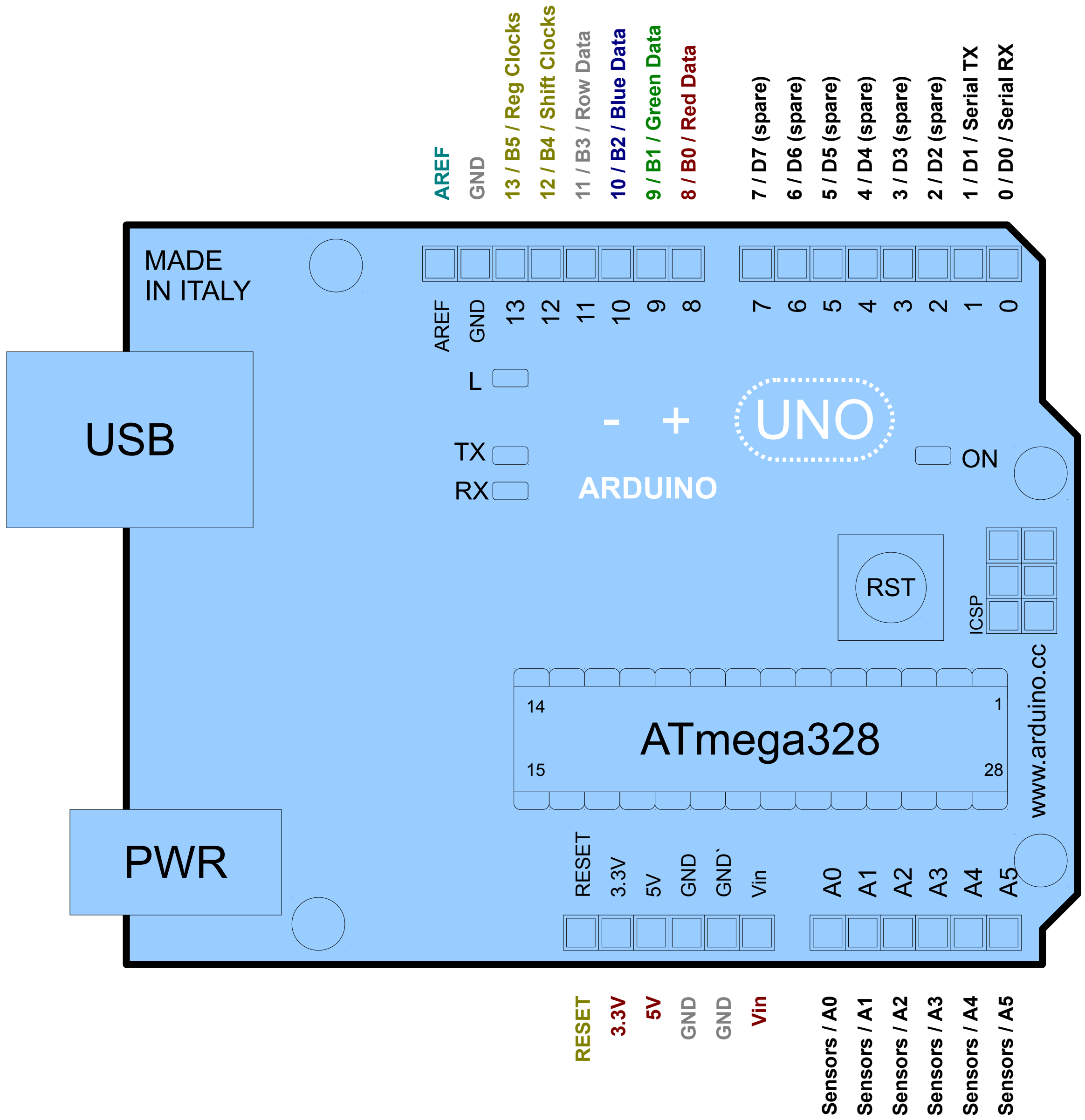
LED Dance Floor Controllers

- Our first design (v1) tried to use a parallel port to clock and fill the shift registers
- This used to be a common way to control electronics in “the good old days”. Parallel ports used to be quite dumb and you could clock any data you wanted out of them
- These days, PC parallel ports (especially USB ones) are much more “intelligent”, they “negotiate” with your printer in interesting ways, they buffer the data, they mess with the timing. Great for printers, terrible for dance floors!

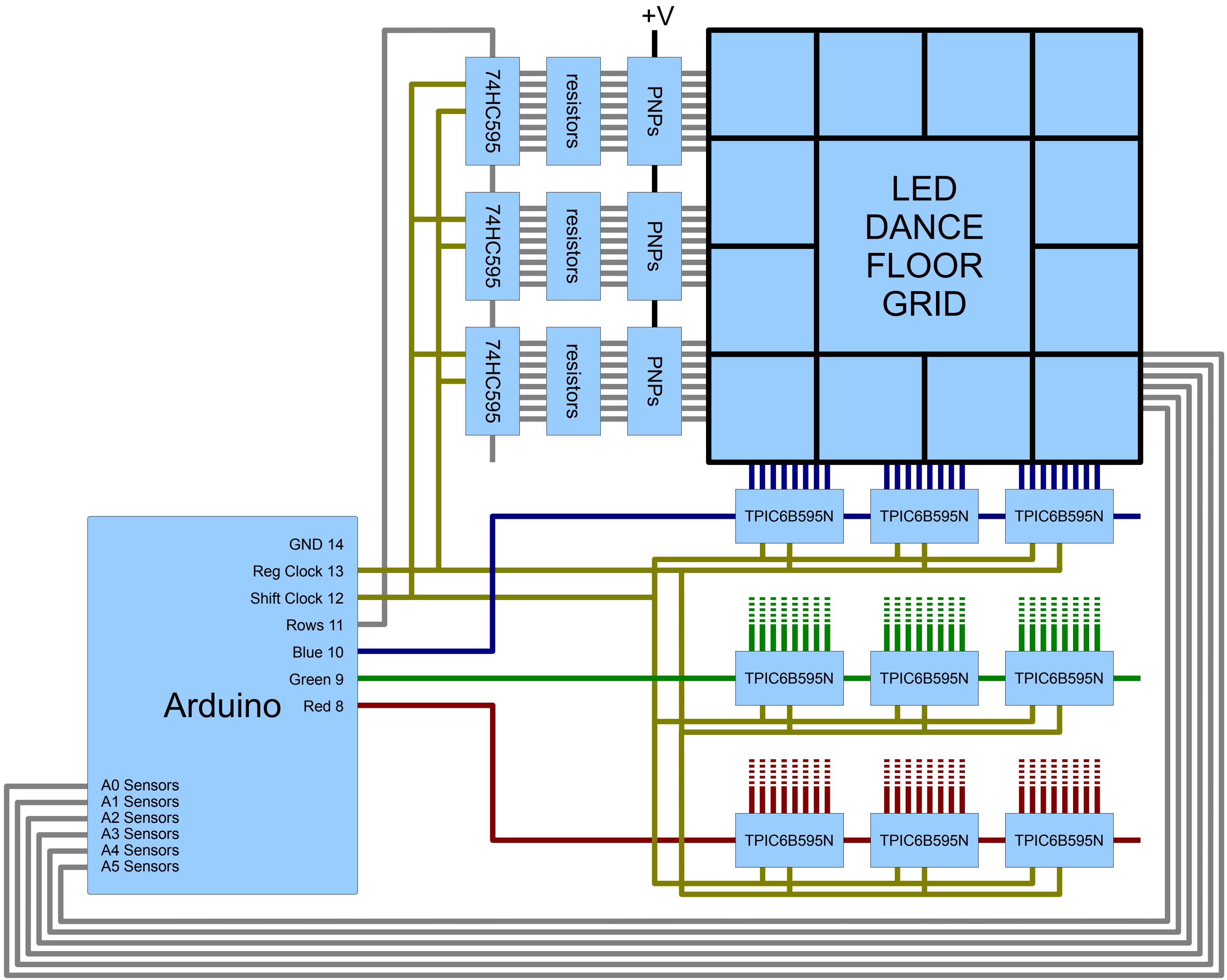
LED Dance Floor Controllers

- We fairly quickly switched to a design (v2) which used the same shift registers, but an Arduino microcontroller instead of a parallel port
- This was a great decision, it has about 20 pins which can be inputs or outputs, instead of the Parallel port's 8-ish kinda half-duplex ones
- It also allows us to program games **on the Arduino**, then remove the laptop/USB and the floor can continue to play, self-contained!

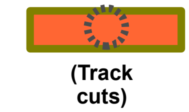
Arduino Signals

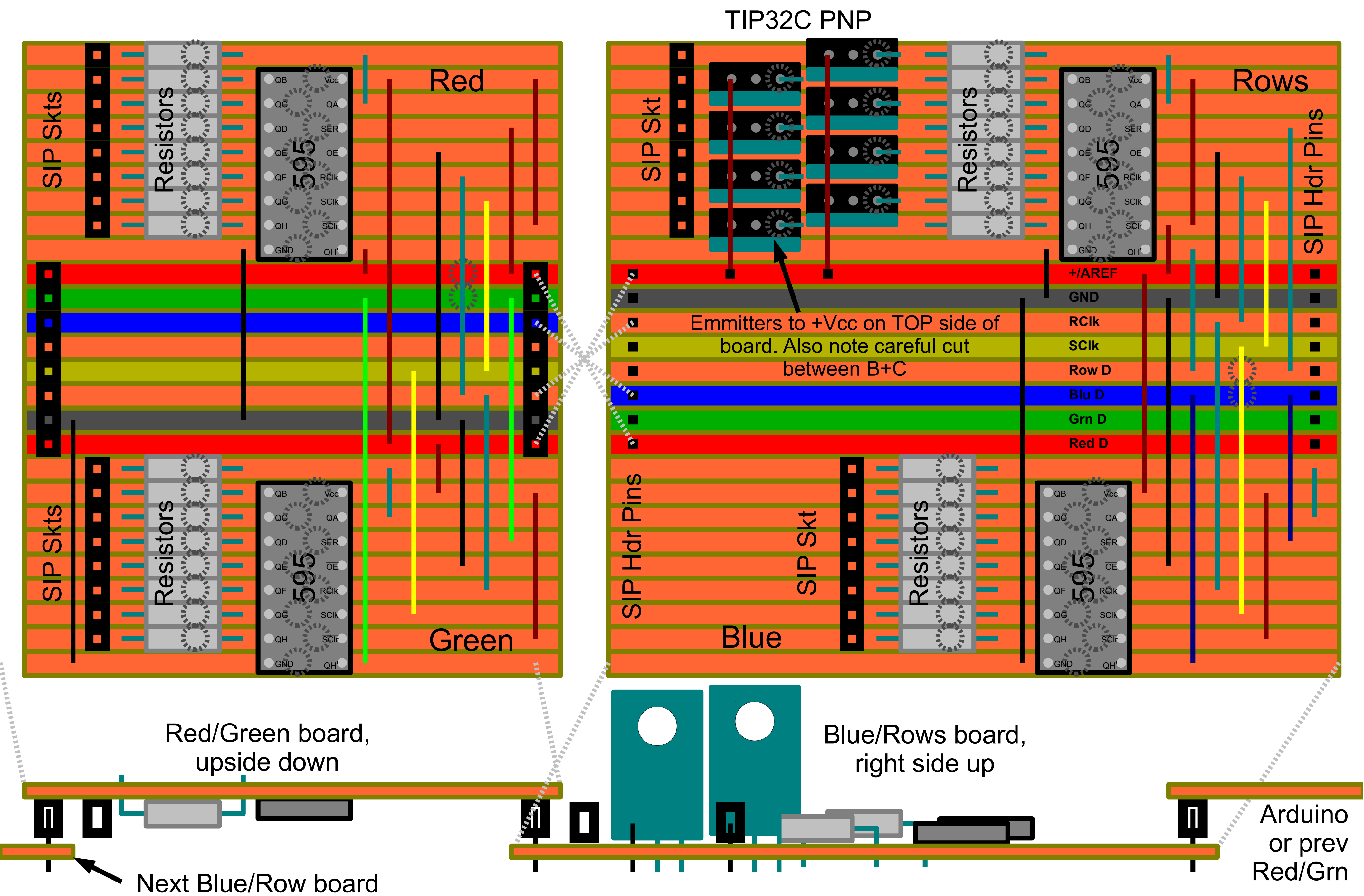


Block Diagram

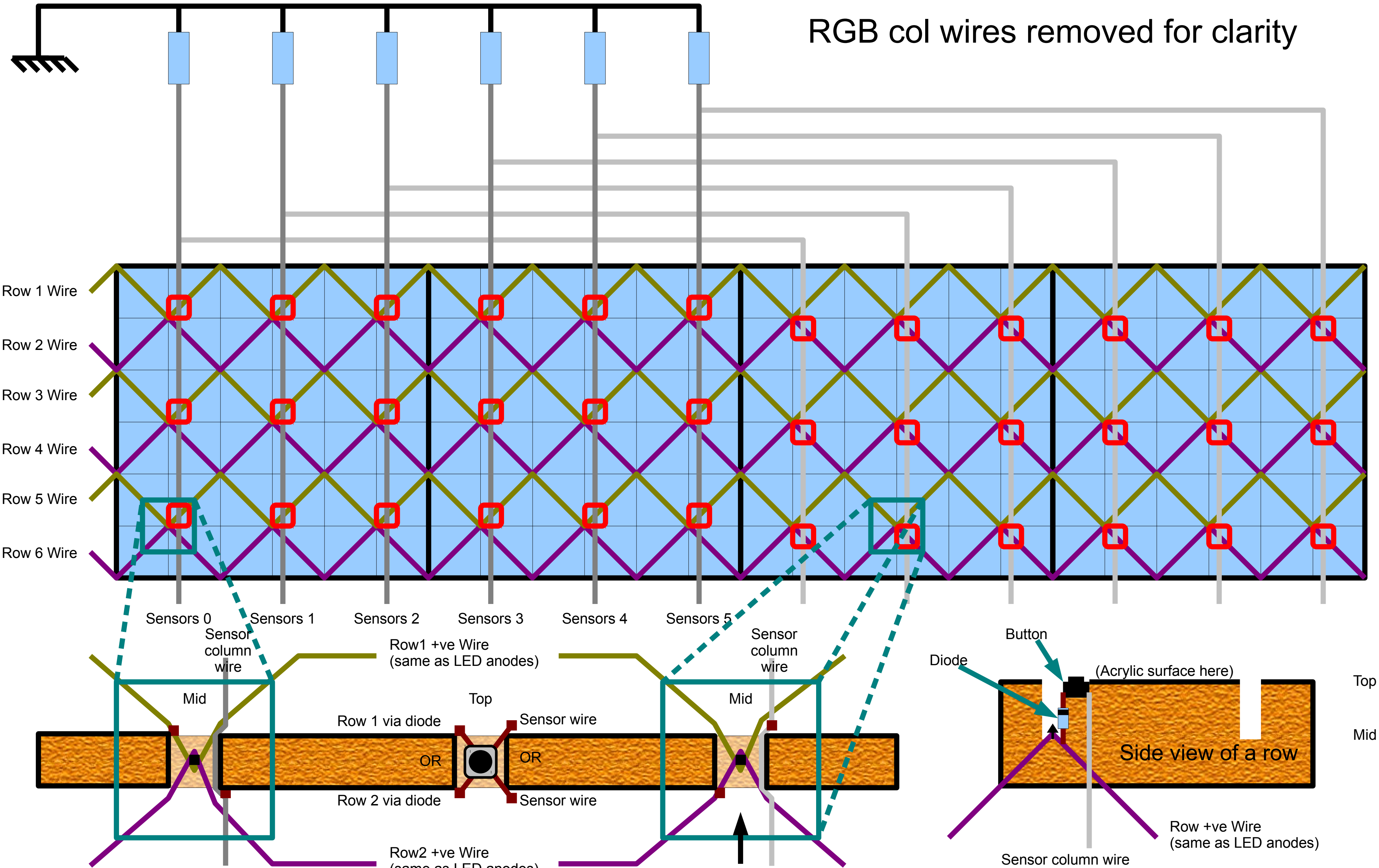


Shift Registers on Vero/Stripboard

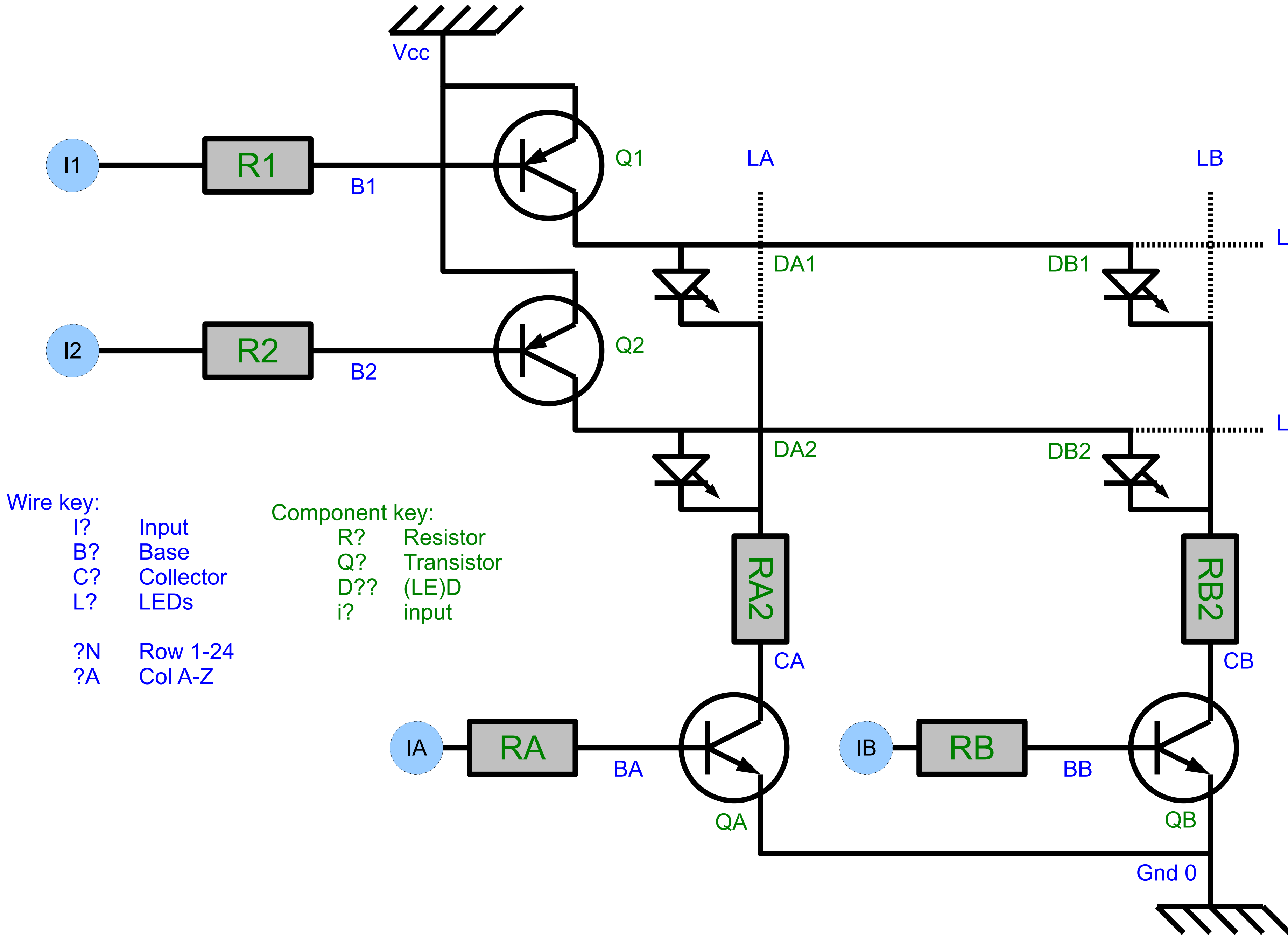




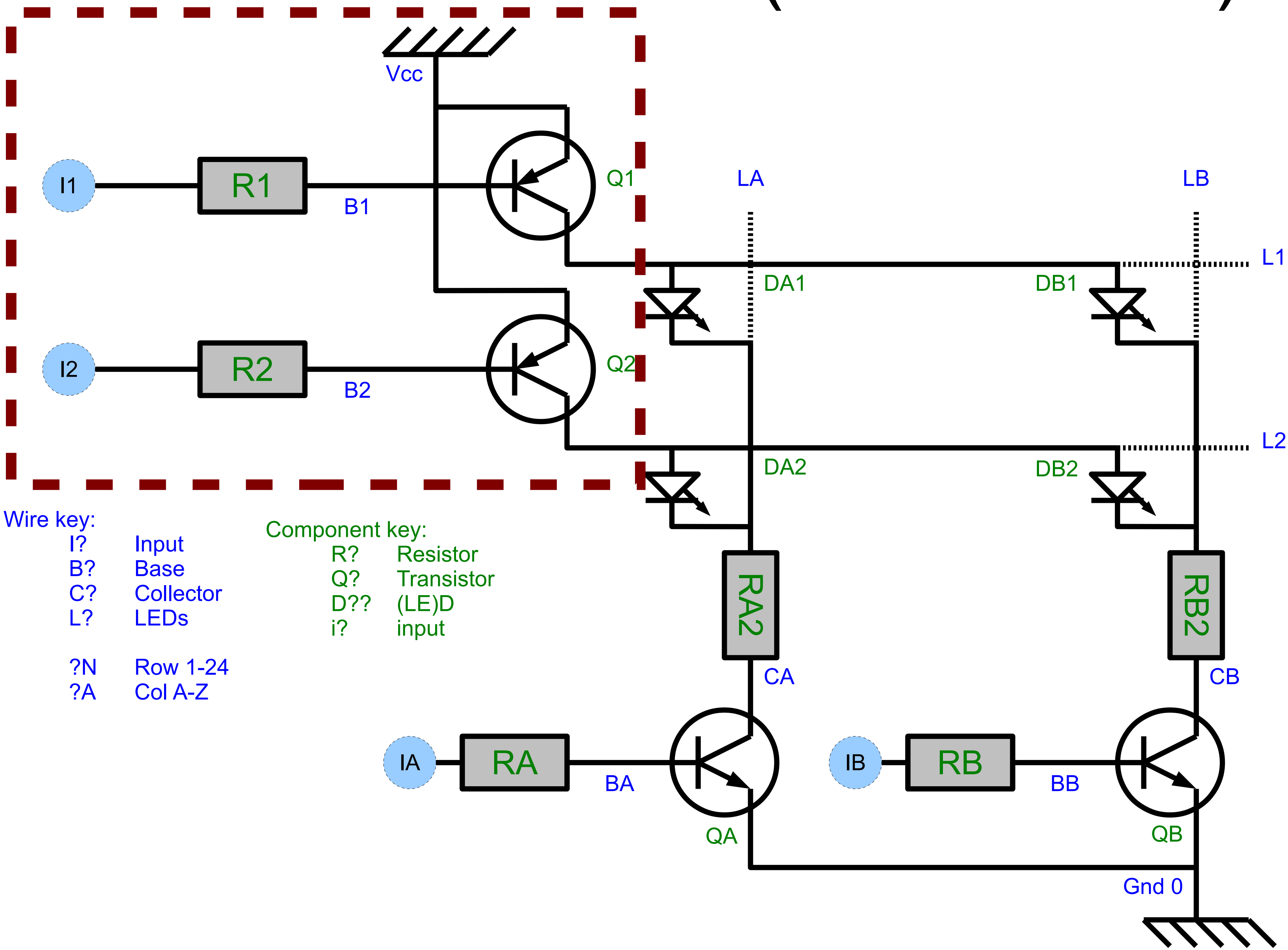
Sensors Detail



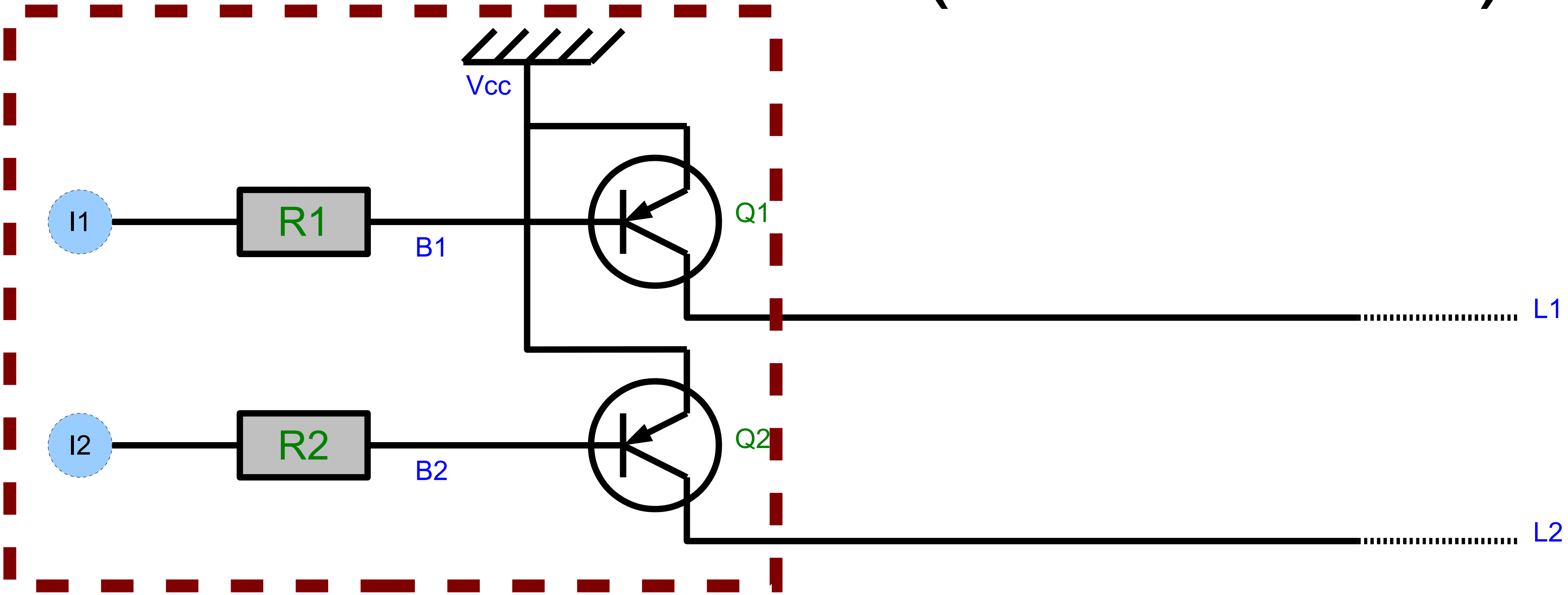
SPICE test circuit (2rows 2cols)



SPICE test circuit (2rows 2cols)

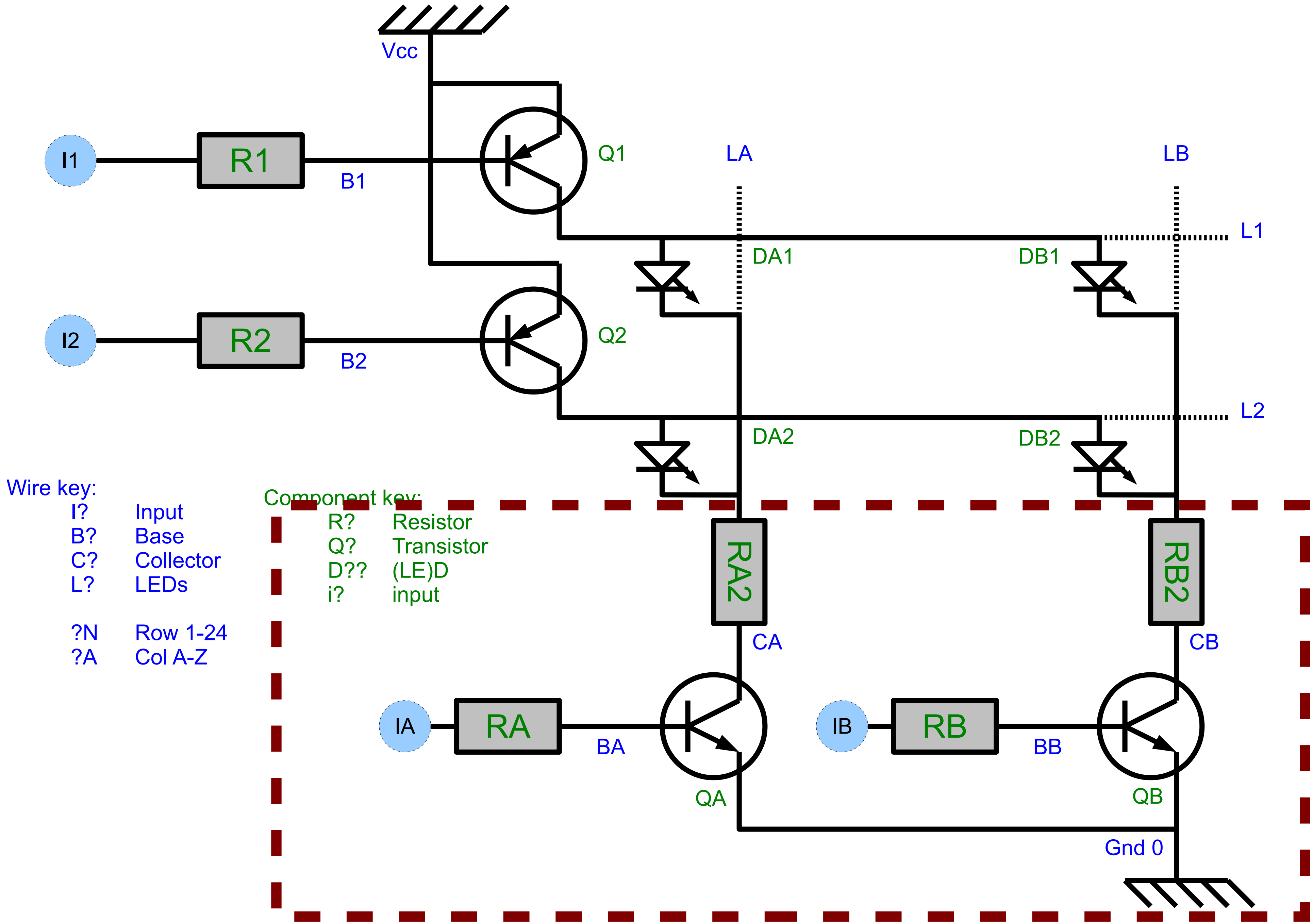


SPICE test circuit (2rows 2cols)



* Comp	N+	N-		Value ...
Vcc	Vcc	0		DC 5V
* Row driver circuitry				
R1	I1	B1		510
R2	I2	B2		510
* Trans	C	B	E	Model
Q1	L1	B1	Vcc	PNP
Q2	L2	B2	Vcc	PNP

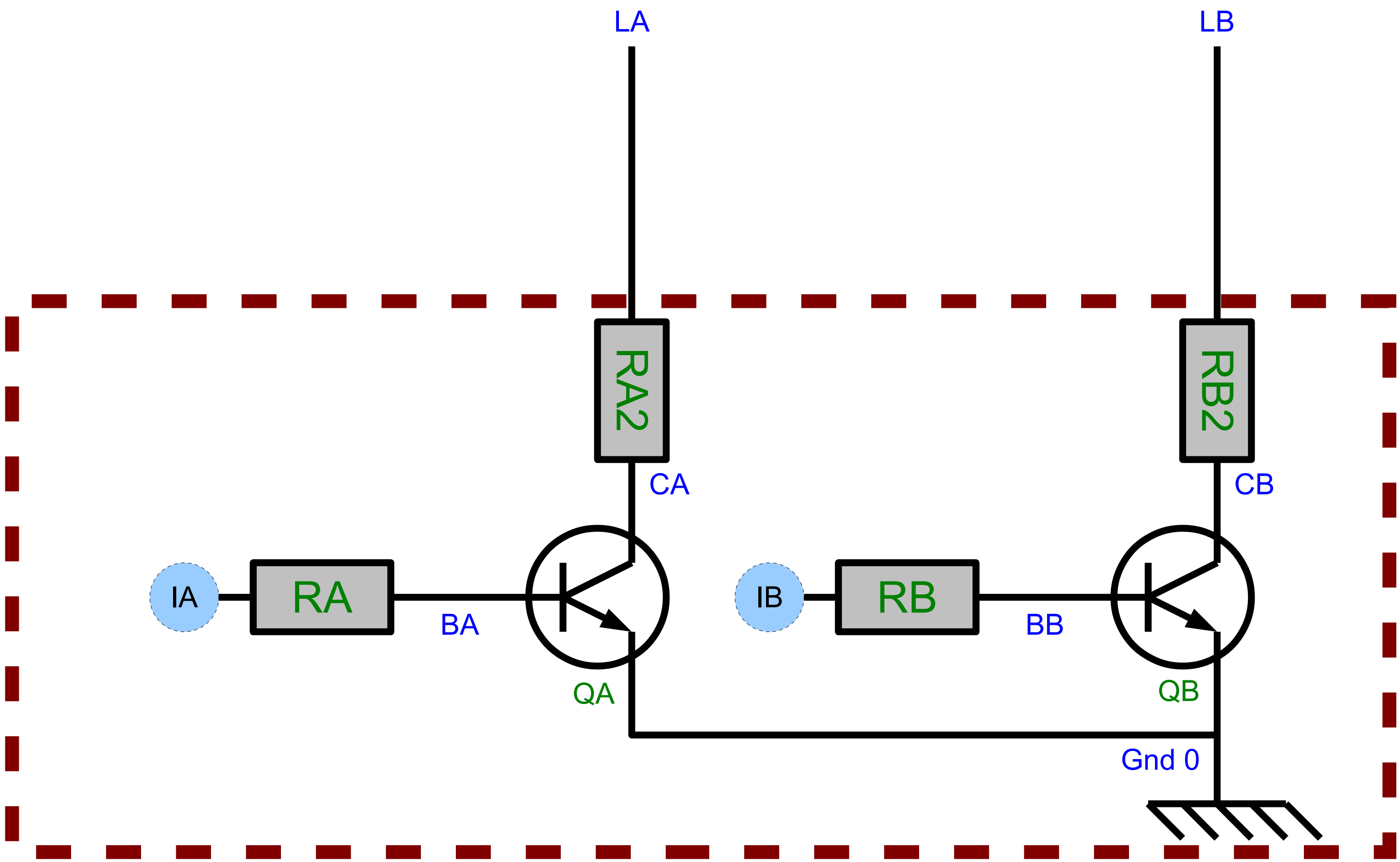
SPICE test circuit (2rows 2cols)



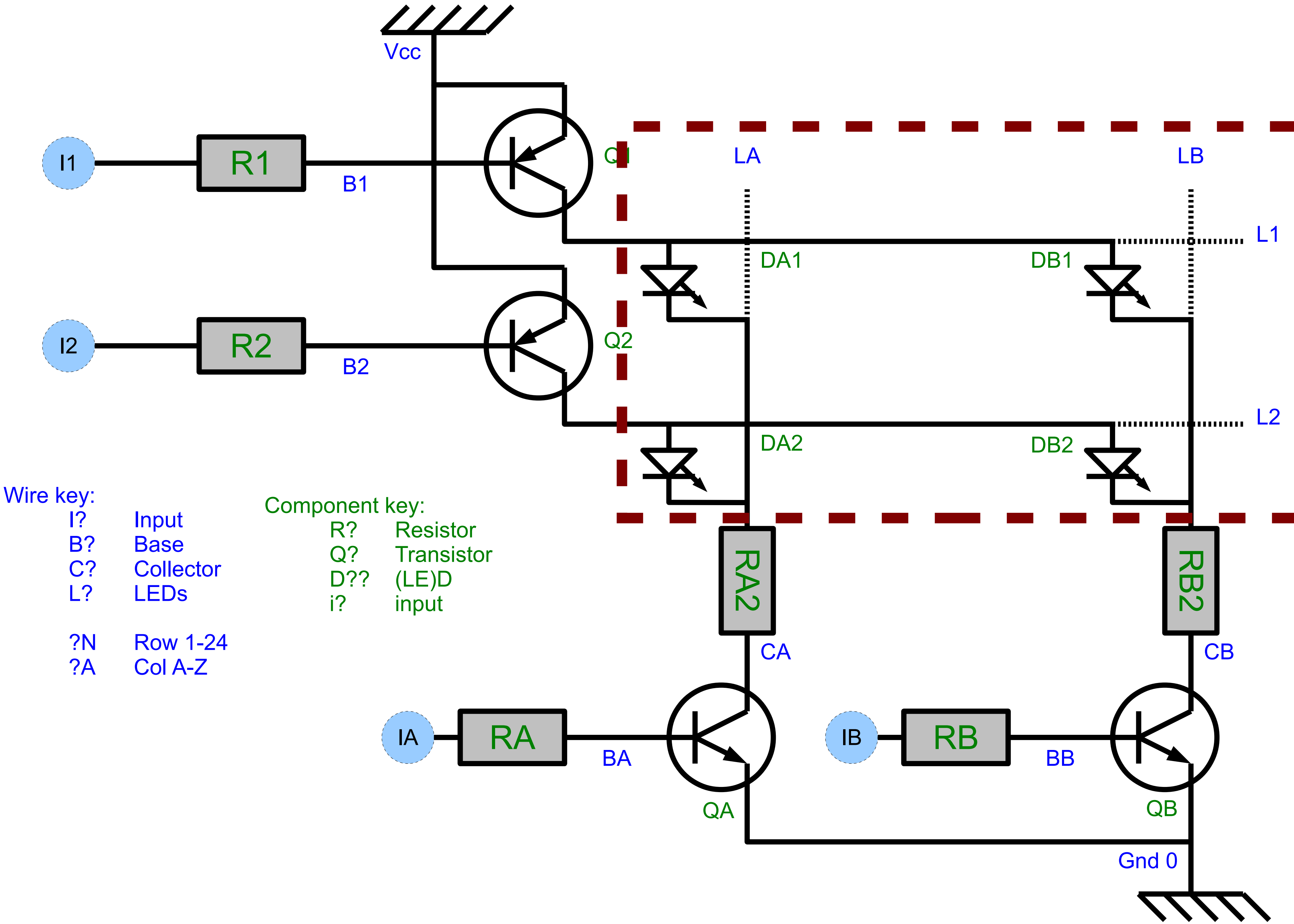
SPICE test circuit (2rows 2cols)

* Column Driver circuitry

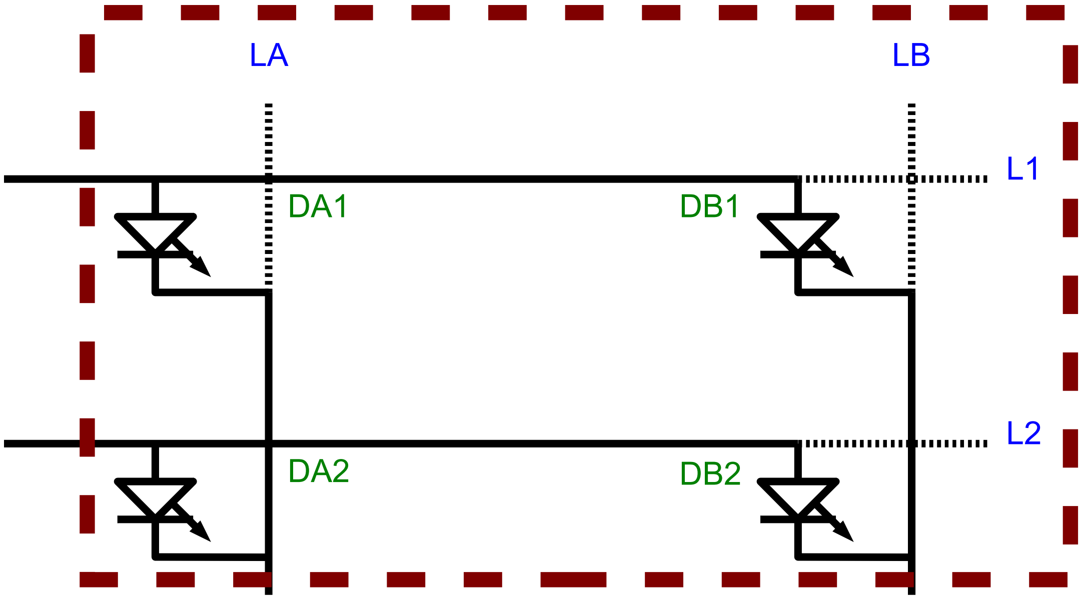
* Comp	N1	N2		Value ...
RA	IA	BA		510
RB	IB	BB		510
RA2	LA	CA		150
RB2	LB	CB		150
* Trans	C	B	E	Model etc
QA	CA	BA	0	NPN
QB	CB	BB	0	NPN



SPICE test circuit (2rows 2cols)



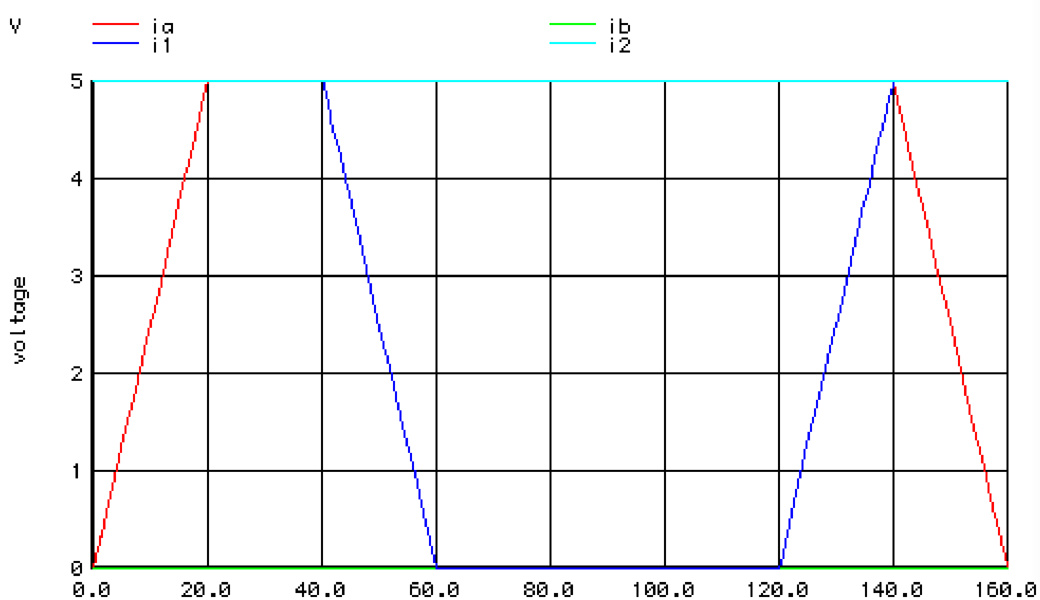
SPICE test circuit (2rows 2cols)



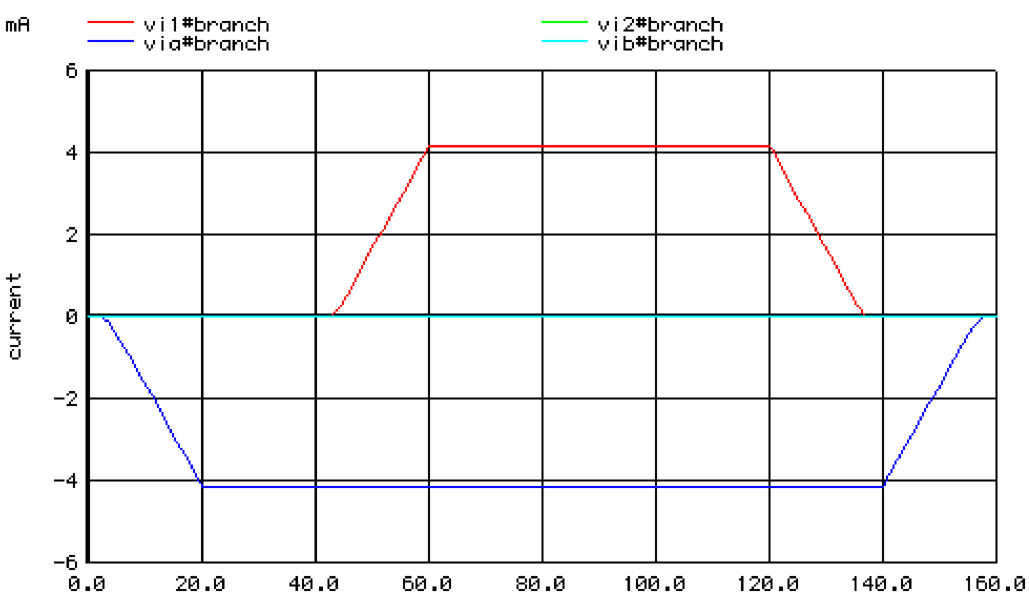
* Diode	N+	N-	Model ...
* LEDs			
DA1	L1	LA	LED
DA2	L2	LA	LED
DB1	L1	LB	LED
DB2	L2	LB	LED

* Yes, there's probably better ways to do this with .SUBCKT stuff
* – maybe when I want to simulate the whole 24x24 :-)

First SPICE results



Our input signals.
0-20µs we select one column
40-60 we activate one row
120-140 we deactivate the row
140-160 we deselect the column

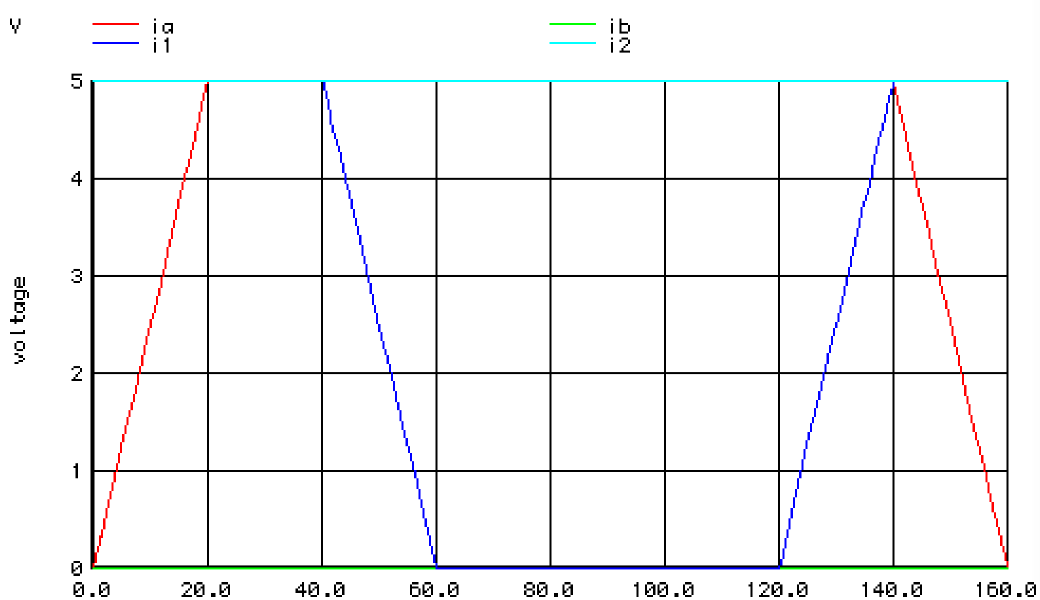


Our input currents.

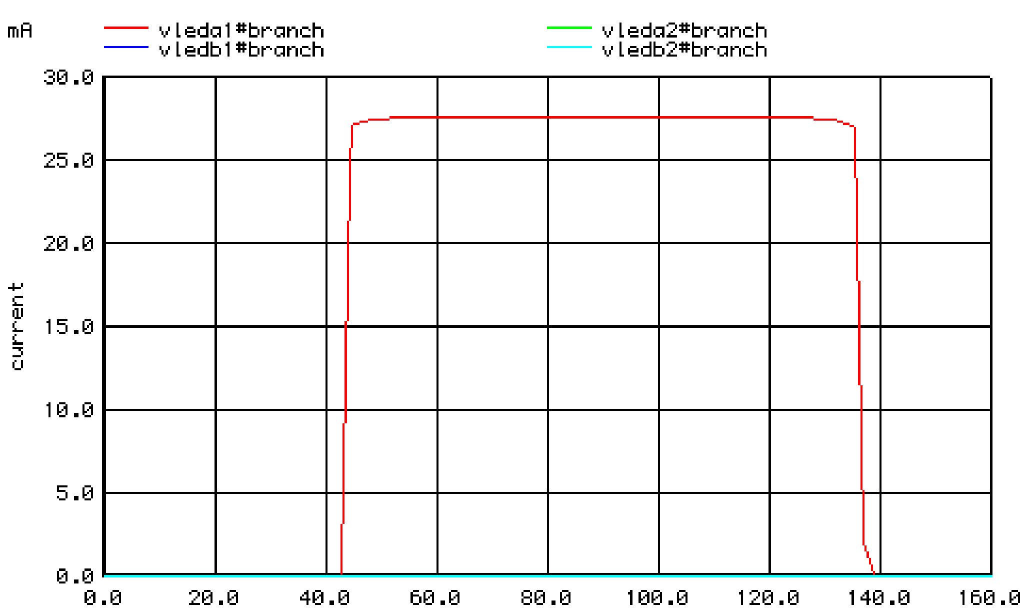
There's a reasonably good reason why one current looks positive (sinking) and one looks negative (sourcing).

These currents would certainly not burn out the shift registers, but could probably be reduced. This was just with a “first guess” of 1kΩ for the base resistors – not too bad!

First SPICE results



Our input signals.
0-20µs we select one column
40-60 we activate one row
120-140 we deactivate the row
140-160 we deselect the column



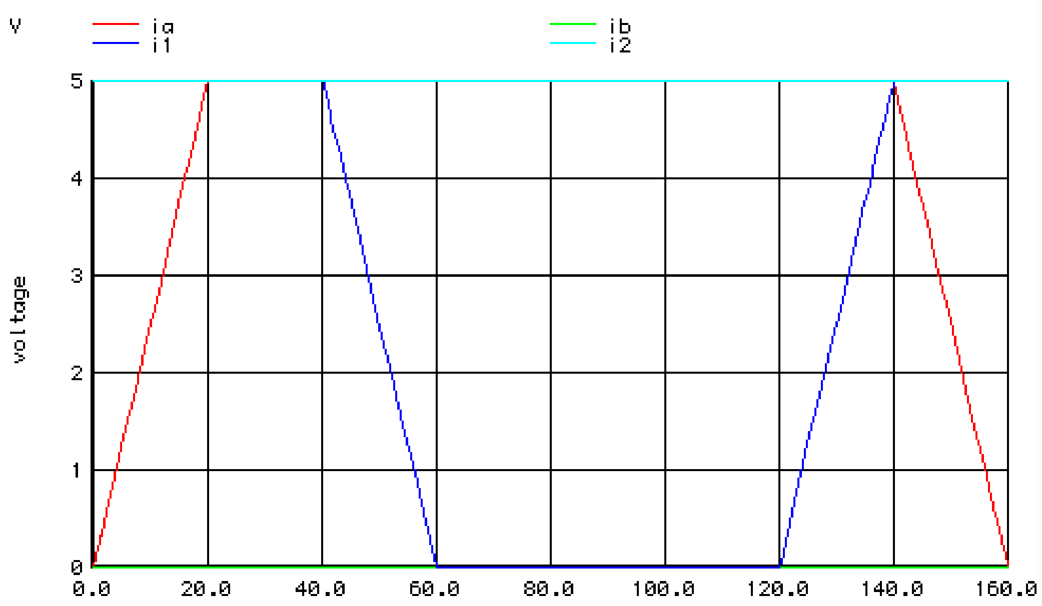
Our LED currents.

As expected, only 1 of the 4 LEDs in our test circuit is lit.

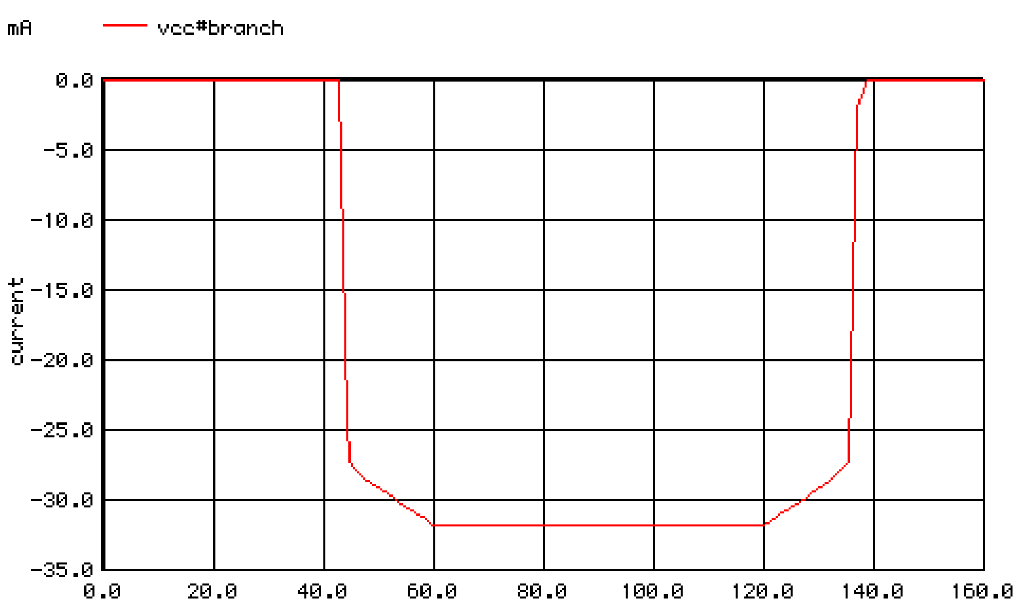
The current looks high, but I'm pretty sure that's because version 1 of my test circuit used a plain 0.7V diode model.

I could do with a more accurate LED model with 2.1V-3.2V forward voltage.

First SPICE results



Our input signals.
0-20µs we select one column
40-60 we activate one row
120-140 we deactivate the row
140-160 we deselect the column



Total supply current.

Again, there's a good reason why this counts as being negative.

We're certainly not short-circuiting anywhere though. Looks pretty believable for a first LED dance floor circuit simulation, and my first attempt to use SPICE for nearly 2 decades!

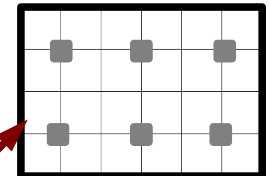
Some game ideas

Idea

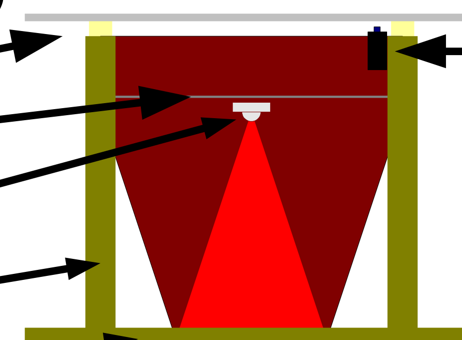
Idea

Dance floor made of 4x6 modules...

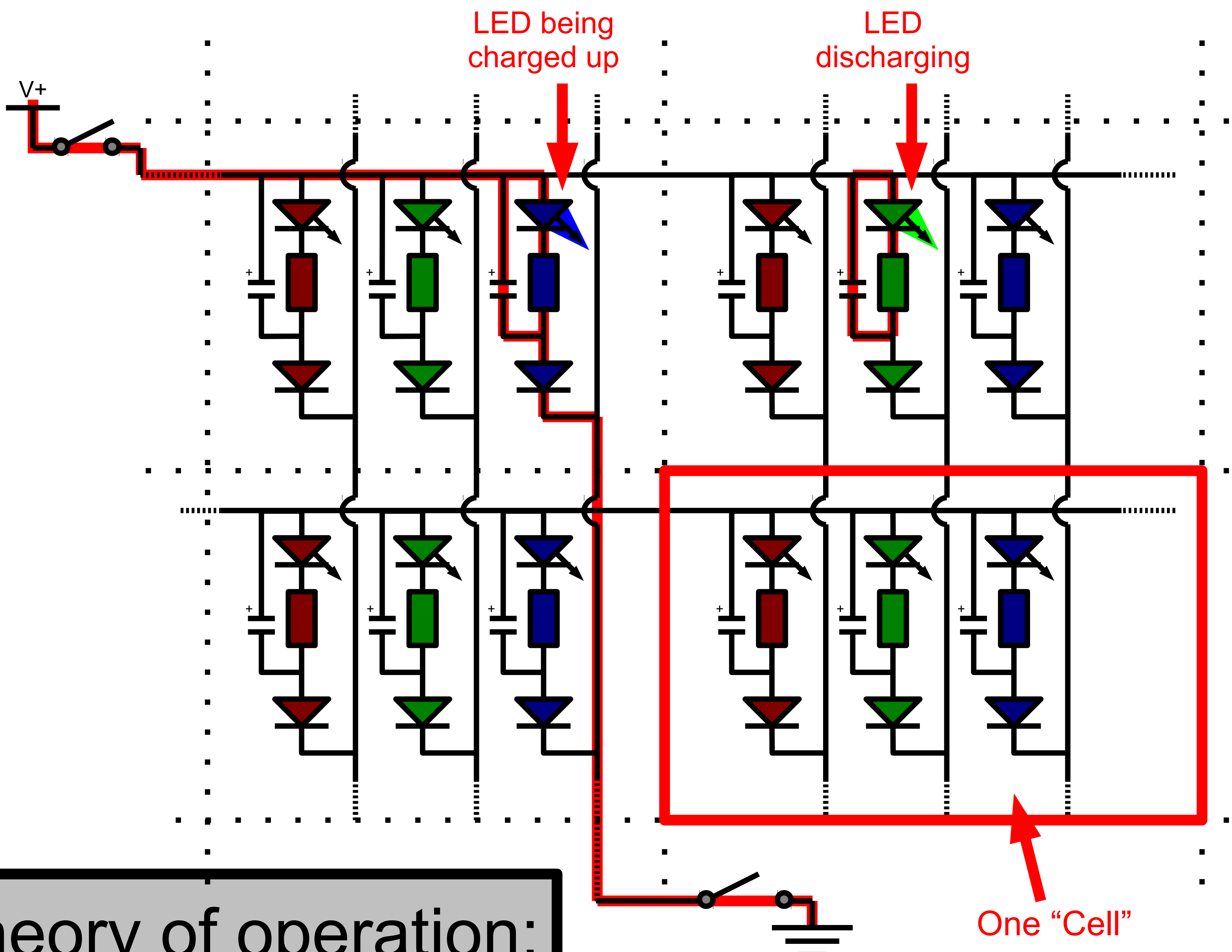
... and a 4-inch border for control electronics and cabling



Each module is 6x4 cells, sized to fit the cheapest acrylic we could find.
Each cell is 4x4 inches, minus “1 inch” (before planing) spruce which makes up the cell walls, slotted into each other.
There are micro-switches in between each set of 2x2 cells to detect players standing/jumping there.

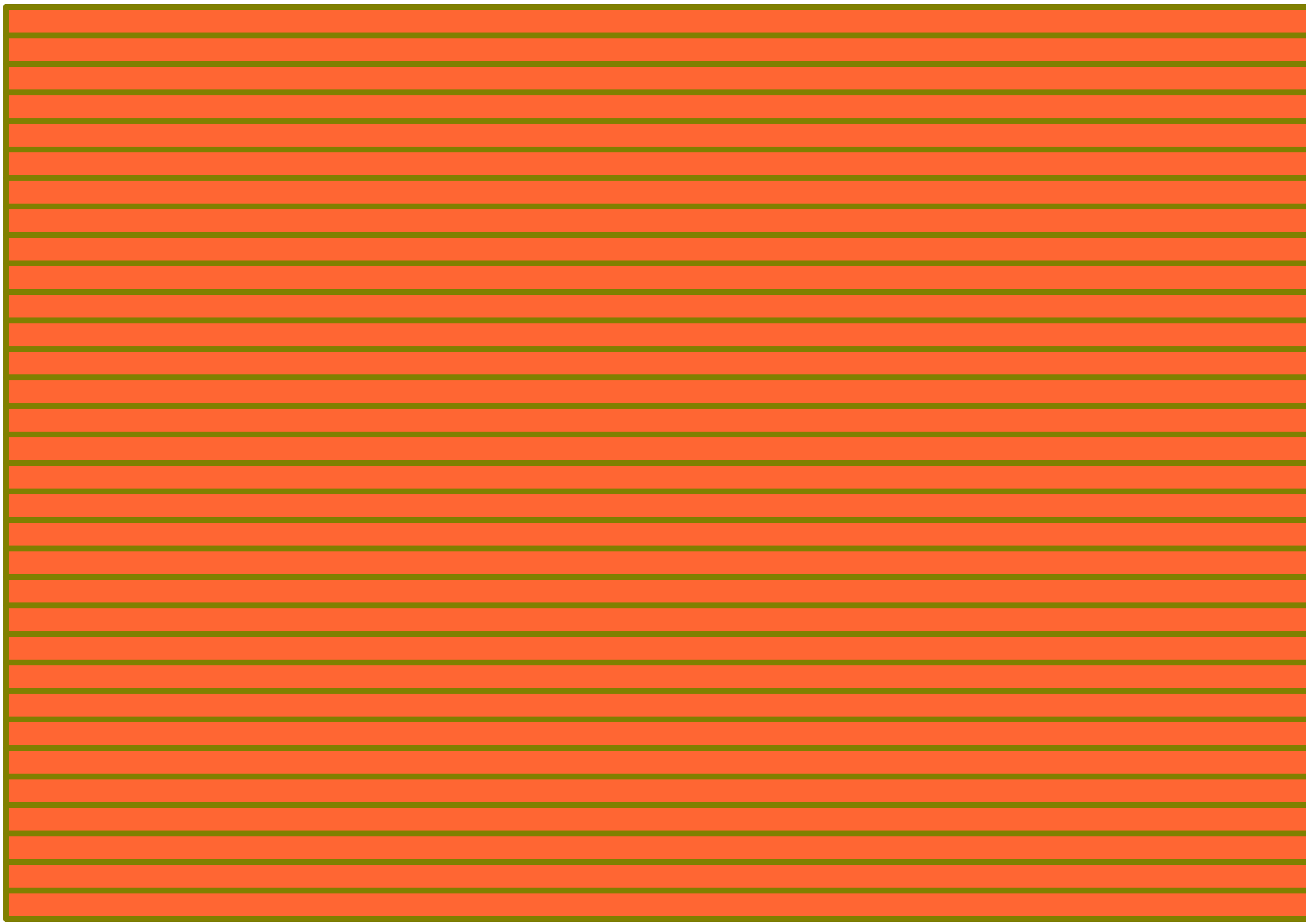


Old 6x4-cell theory

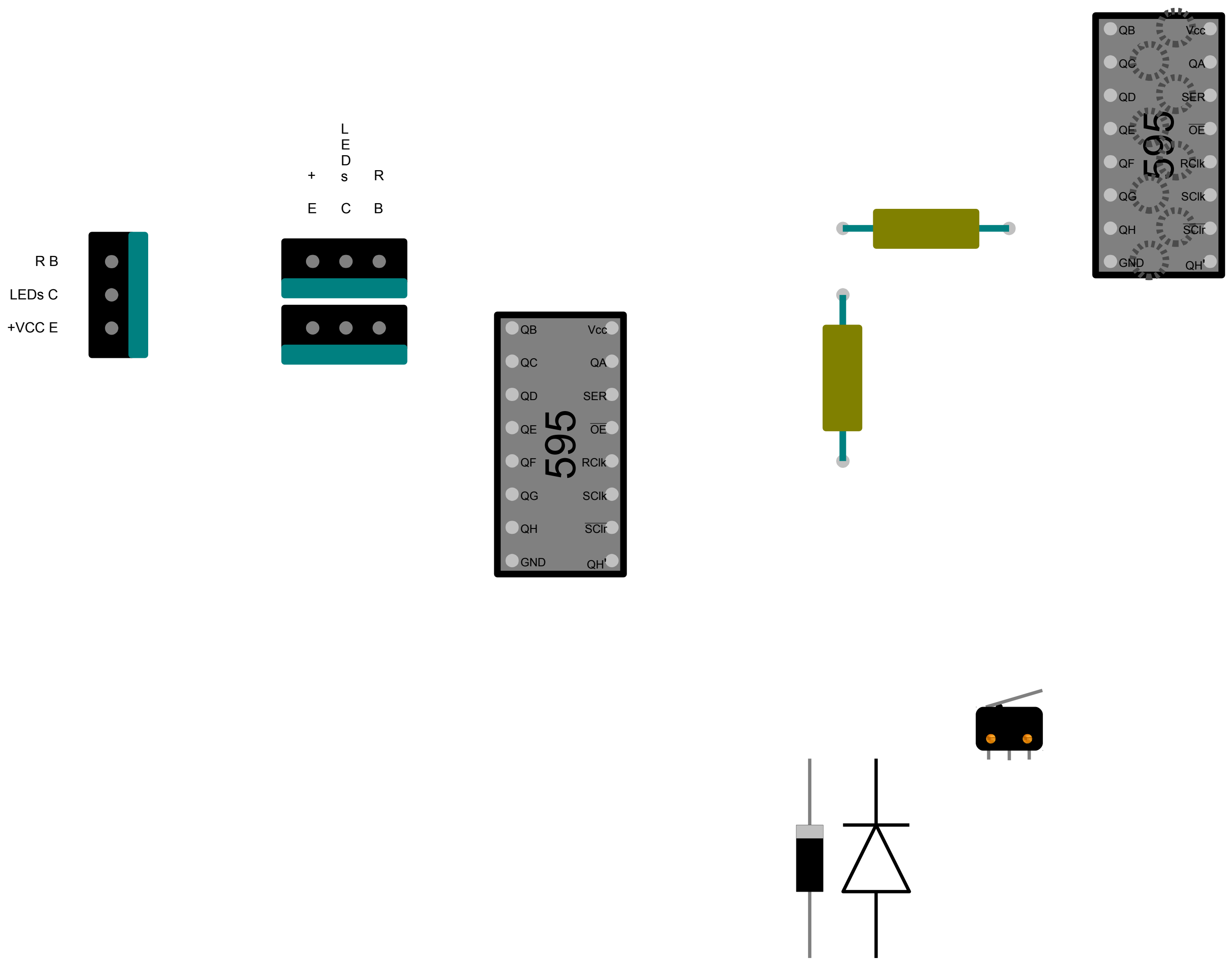


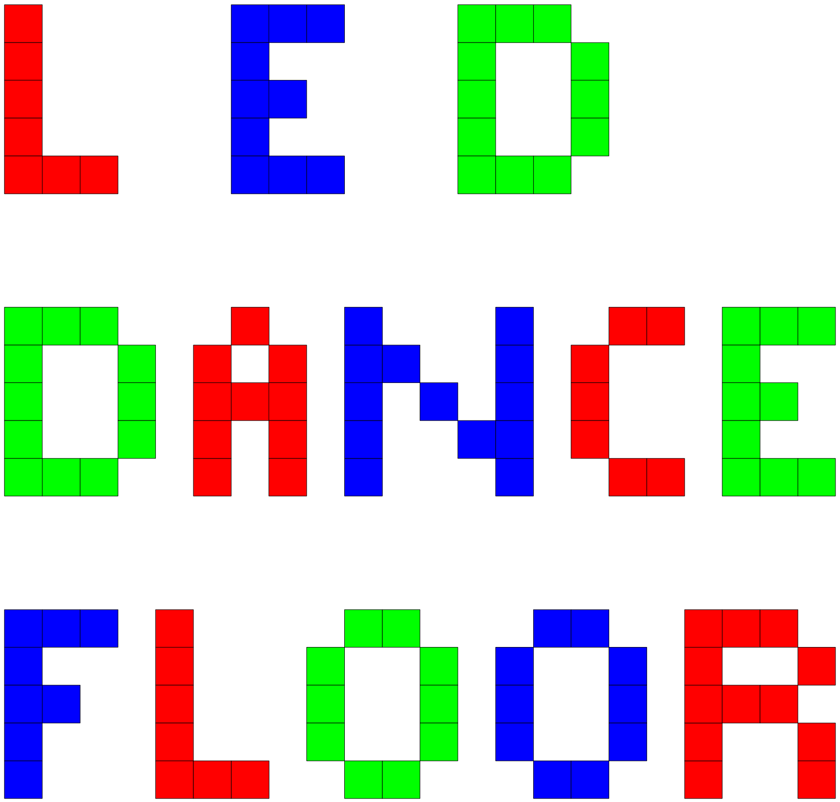
Theory of operation: brighter version

Veroboard



Components


R B
LEDs C
+VCC E
+
LEDs
R
E
C
B
QB
Vcc
QC
QA
QD
SER
QE
OE
QF
RClk
QG
SClk
QH
SClr
GND
QH'
595
QB
Vcc
QC
QA
QD
SER
QE
OE
QF
RClk
QG
SClk
QH
SClr
GND
QH'
595


LED
DANCE
FLOOR