May 29, 2013
PROTO BOARD GENERATION: VERSION 1

1. Introduction

In this document, I describe my first attempt at solving the proto board problem – to make it easier to understand the code written in the package `https://github.com/mikemeko/6.01_Tools/tree/master/src/circuit_simulator/proto_board`.

2. Terms

2.1. **Circuit.** Describes a schematic circuit as would be drawn on the GUI board by a user. An example is given in Figure 1.

2.2. **Circuit Piece.** (Abbreviated *piece*) Describes a piece of hardware that might be put on the proto board. Examples are Op Amp packages and resistors as shown in Figure 2.
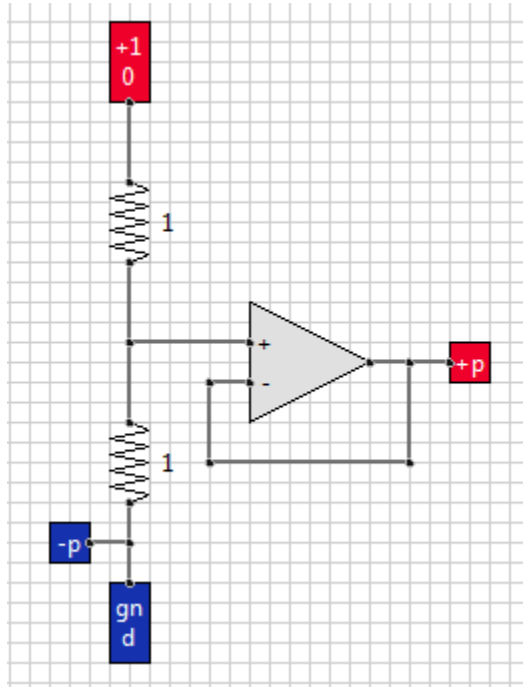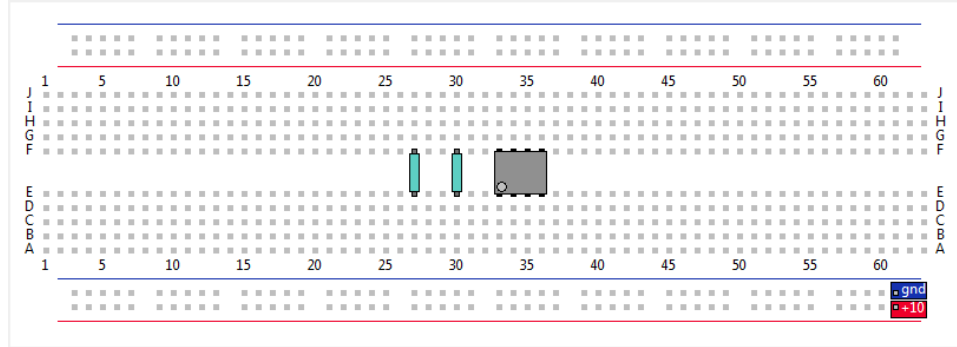

FIGURE 1. A voltage divider *circuit*.

FIGURE 2. Examples of *circuit pieces*.

FIGURE 3. *Circuit piece placement* for the voltage divider circuit.

[Related code: `https://github.com/mikemeko/6.01_Tools/blob/master/src/circuit_simulator/proto_board/circuit_pieces.py`]

2.3. **Circuit Piece Placement.** (Abbreviated *placement*) Describes a placement of a number of circuit pieces on the proto board, **with no wiring**. Figure 3 shows an example of a placement for the voltage divider circuit shown in Figure 1.

2.4. **Location.** Indicates a point/dot on the proto board that may be described by row and column, for example.

3. STEPS

3.1. **Circuit → Placement.** The first step is to come up with an appropriate placement for a given circuit. An example of this is the transition from Figure 1 to Figure 3. Finding a placement from a circuit involves trying among various options:

(1) There are multiple ways to package together Op Amps. For instance, if there are two Op Amps in the circuit, we can either put them together in one Op Amp piece, or we can use two Op Amp pieces (one for each Op Amp). In my solution, I try all possible ways of packaging together the Op Amps in the circuit and choose the "best" one among all.

(2) Some pieces (like Op Amp packages and resistors) can only be placed in the middle strip of the proto board, but others (like the connectors and pots) can be placed at various vertical locations on the proto board. For these pieces, we try all possible ways of placing them on the proto board (e.g. for pots, top half or bottom half) and choose the "best" one among all.

(3) The orientation of some pieces (like Op Amp packages and resistor) affects the resulting wiring. We try all possible ways of orienting these pieces and choose the "best" one among all.

(4) Given a set of pieces, what remains to decide is the ordering of the pieces. The ideal thing to do is to try all possible permutations of the pieces, but this explodes quickly. My solution here is placing the pieces on the board one at a time by inserting the kth ($k \geq 1$) piece in the best place among the k possible places it can be put while preserving the order of the other $k-1$ pieces.

Now, a natural question to ask is: how do we choose the "best" of set of placements? This is done by assigning a cost to each of the placements and choosing the placement with the minimum cost. The cost for a placement is evaluated by first finding all the connections that have to be made, and then computing the sum of the Manhattan distance for each of the connections.

Another natural question to ask is: how do we find the connections that need to be made given a placement? We focus on each of the nodes in the circuit, one at a time. Given a node, we can find all of the locations on the proto board that are a part of that node. What we need to do is put wires on the proto board so that these locations are connected with one another. As the "connections" to be made for the node, the current solution chooses pairs of these locations for the node that essentially produce a minimum spanning tree of the graph of locations, where the distance between two locations is simply the Manhattan distance. This is done for each of the nodes in the circuit to produce one large collection of connections to make.

[Related code: `https://github.com/mikemeko/6.01_Tools/blob/master/src/circuit_simulator/proto_board/circuit_to_circuit_pieces.py`, `https://github.com/mikemeko/6.01_Tools/blob/master/src/circuit_simulator/proto_board/circuit_piece_placement.py`]

3.2. **Placement → Wired Proto Board.** Once an appropriate placement is chosen, we carry out the wiring. We do this by connecting one pair of locations on the proto board at a time. While there are pairs of locations remaining in the list of pairs of locations to connect, we apply the following steps:
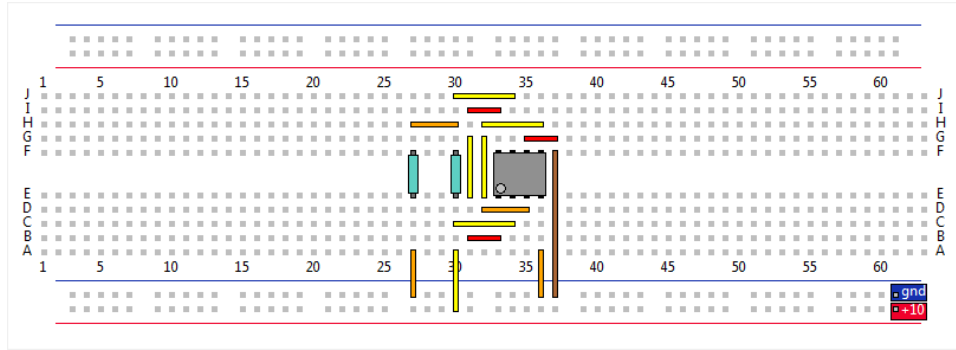
FIGURE 4. A wired proto board for the voltage divider circuit.

(1) Condense: we go through the list of pairs of locations and we try to replace each by another (equivalent) pair of locations that is closer together.
(2) Pick next pair: we then chose the next pair of locations to connect. We chose the pair of locations that are farthest away from each other. This decision was reached experimentally. One explanation for why this strategy seems to work well is that the farthest away pairs of locations are in a way the hardest, so it is best to handle them early while there is more space on the proto board.
(3) Connect: once a pair of locations is chosen, the pair is connected by wires. This is done by a search mechanism with various heuristics that favor quickly finding a wiring and making the resulting wiring aesthetically pleasing.
    [Related code: `https://github.com/mikemeko/6.01_Tools/blob/master/src/circuit_simulator/proto_board/find_proto_board_wiring.py`]

An example of a wired proto board is given in Figure 4.