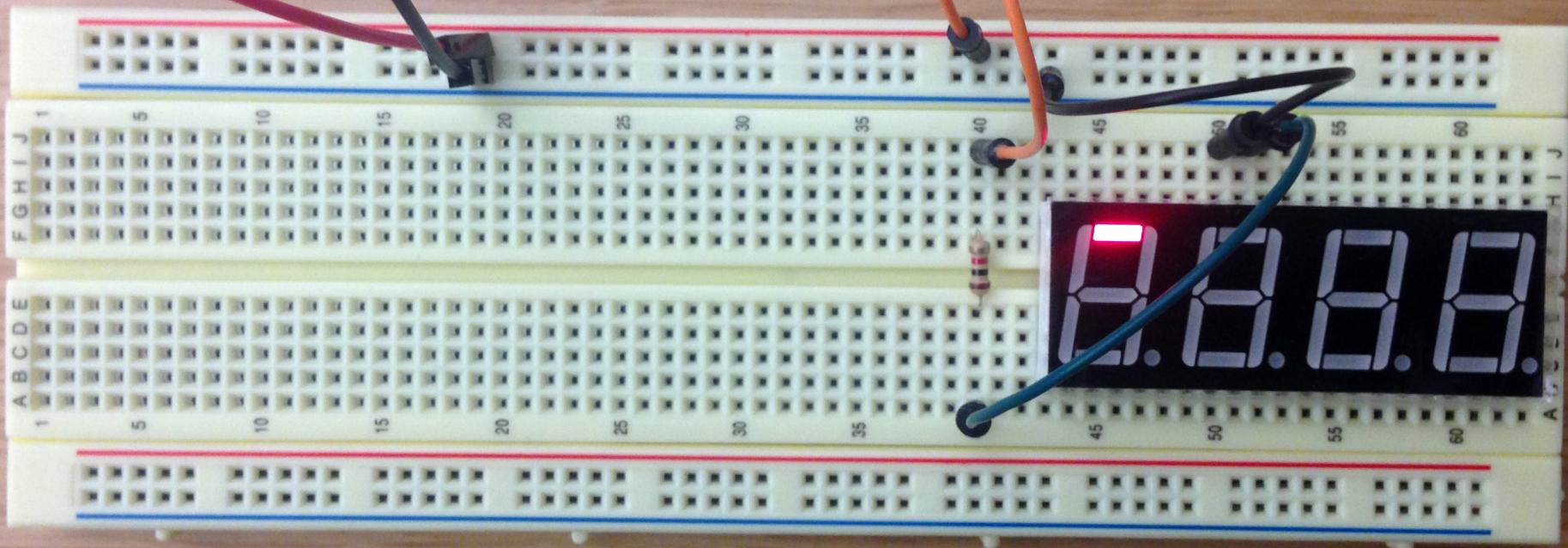
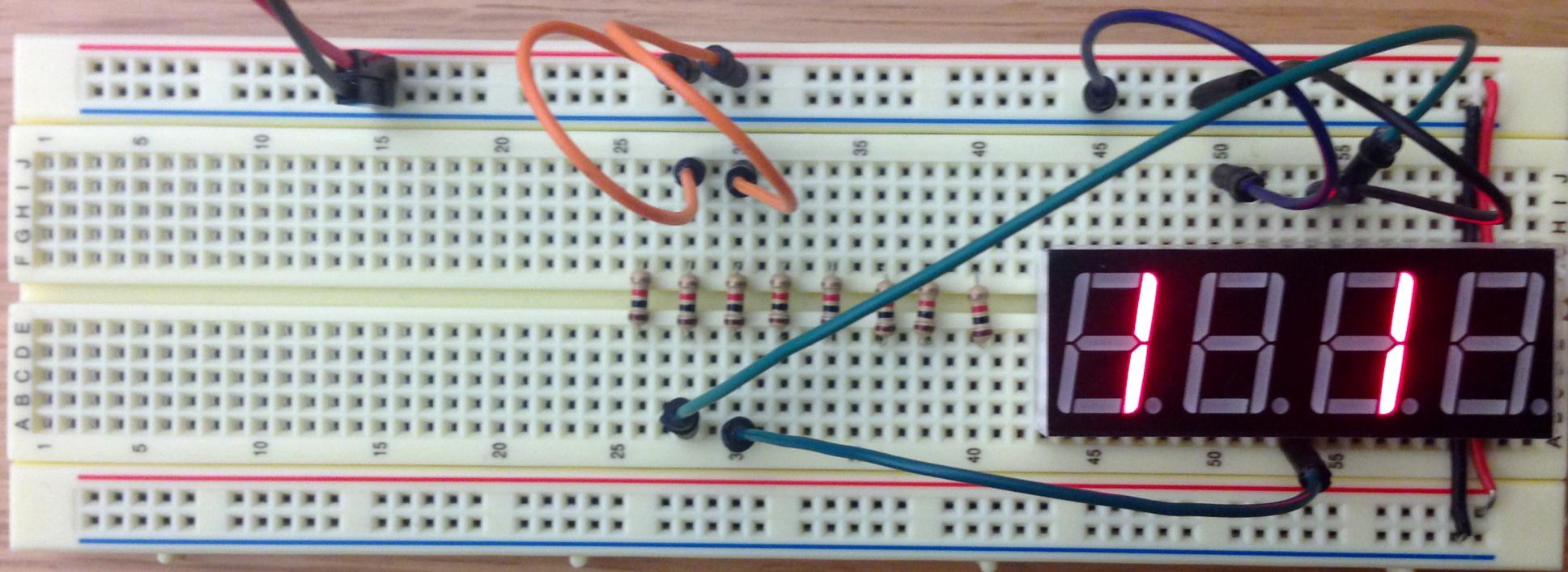
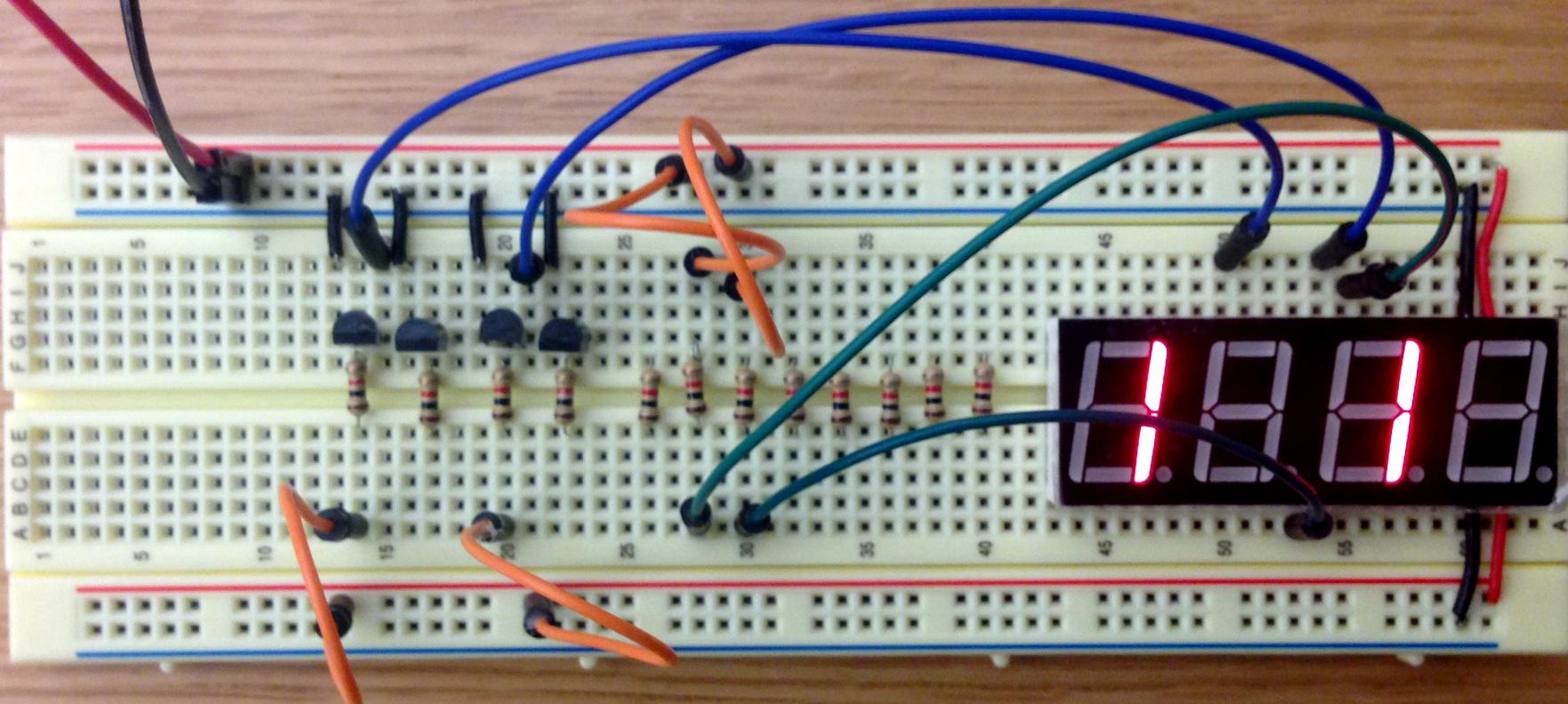


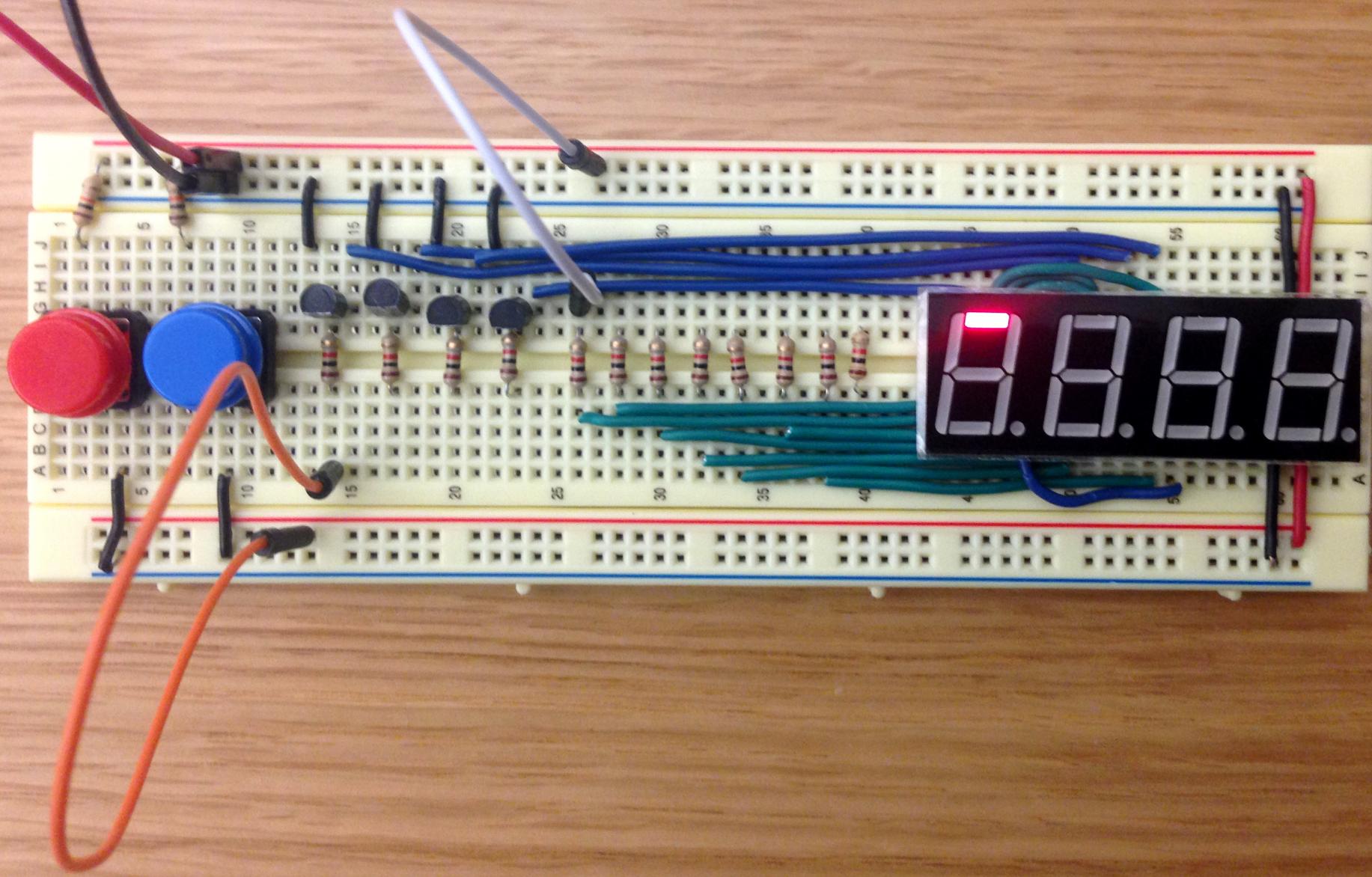
Testing

Test as You Go!









Biggest Rookie Mistake

The worse thing to do is to write a lot of code without testing parts of it.

It will almost certainly not work.

Trust me. I've been writing code that doesn't work for 30 years.

Test Driven Development

Where to Start?

What should you do first?

Implement the simplest testable function first.

Make sure the first function doesn't rely on anything else working.

What Next?

Figure out the best order to develop the rest of your functions. What is a good order? The order that allows you to test as you go!

Corollary: Take baby steps from a known working state to a new working state.

When you add something, rerun all your old tests. It is easy to break something that used to work when you add something new.

Tests are Your Friend

Think of the assertions as a specification of what a function should do. Assertions will clarify your understanding of how things should work.

Implement getters and setters. Test that they work as a matched set (`get(set(x)) == x`).

More generally, develop interfaces that are easy to test.

It often takes thought to figure out how to test certain functions (e.g. `timer_get_time()`).

Tests are Your Friend

Part of the skill of software development is anticipating what can go wrong. We will give you some starter tests, but you are responsible for more thoroughly testing your software.

Trick due to Don Knuth: Develop "torture tests." The border cases, the bizarre cases. What would some maniacal person try to do to trip you up!?

Develop unit tests for each module that are independent of other modules.

`unittest/`