

HARDWARE OF THE COORDINATE DECODER

Probably the biggest design decision is what sort of display to use. There seemed to be a couple of obvious choices: 7-segment LED displays or a small LCD matrix. The former appealed more: they're available cheaply from China via eBay, and they give the gadget a pleasingly retro feel. I might have felt differently had I lived further west: 7-segment displays are only wide enough to display a single 'U', not 'W'.

The display needs to be big enough to display about 32-bits of input data, and about 15 characters of coordinate output. Eight digits seems sensible: 8 hexadecimal digits are exactly 32 bits; and if we display the final northing and easting separately then they'll fit into 8 characters. Octal 7-segment displays are rare, but quad displays are widely available, cheap, and don't require much more wiring.

It would be nice to drive the displays from the microcontroller without any other driver hardware (save current limiting resistors). Counting the decimal-point, each display contains 8 LEDs, and there are 8 such displays. An 8×8 multiplexed design will need 16 output lines on the microcontroller.

Adding a couple of inputs for a rotary-encoder and one input for a button, that makes 19 I/O pins. Given that we'll need at least two power pins, a 20-pin device won't be large enough.

In the end, I picked a 28-pin PIC16F886 mainly because I had some to hand. It's also used by Microchip in one of their demo boards. There might be cheaper options, but I didn't spend time investigating them.

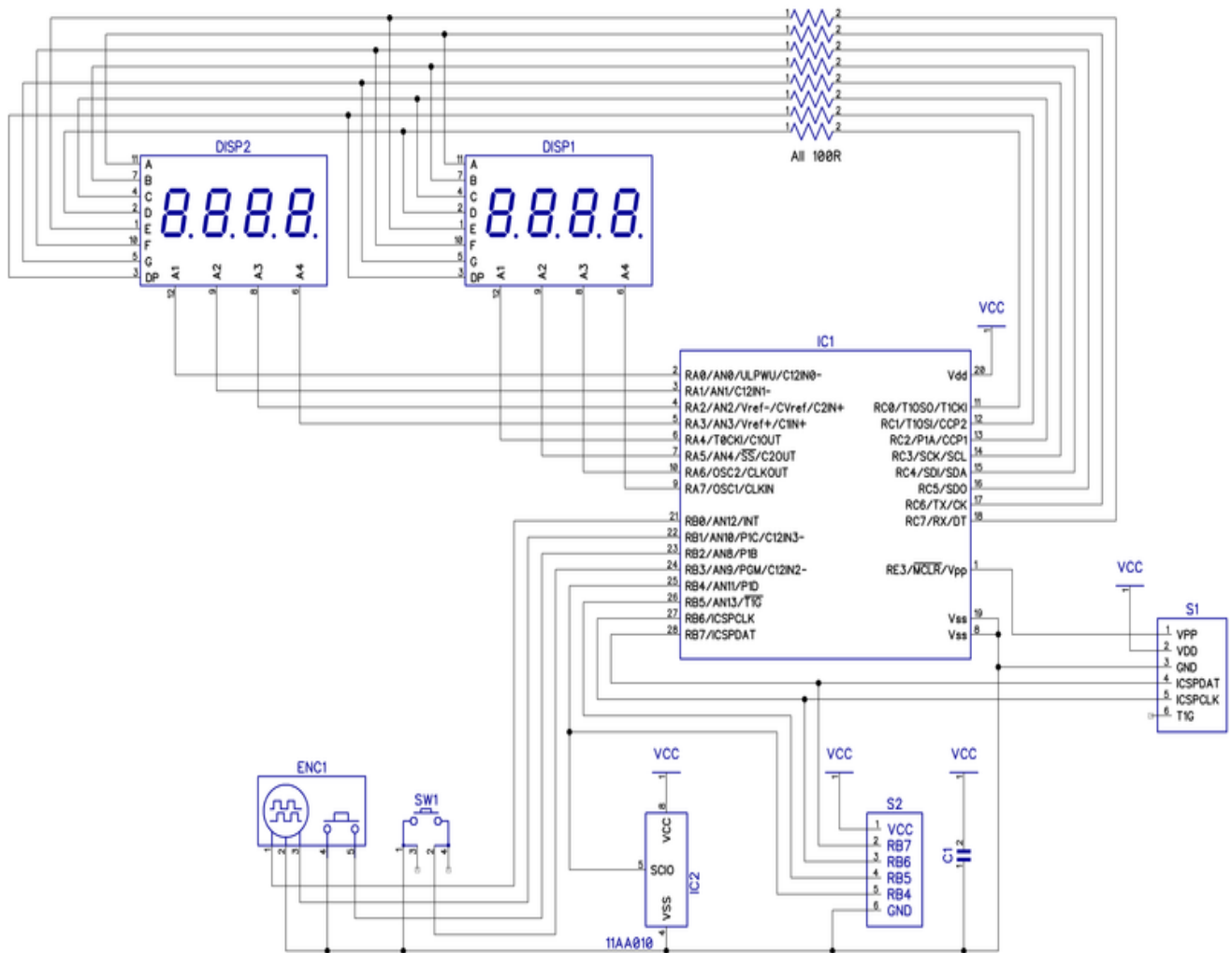
Given that there are some spare I/O pins, it seemed only natural to add a couple of extras:

- a second push-button, integrated into the rotary encoder;
- a UNI/O flash chip in which e.g. the decoder keys could optionally be stored.

Like many PICs the 16F886 has an optional internal 8MHz clock which obviates the need for any external clock circuitry: we don't need particularly high accuracy or vast performance, and power is plentiful.

Happily this still leaves all of the ICSP pins free to facilitate programming.

Here's the final design courtesy of DipTrace:



Source: <http://www.mjoldfield.com/atelier/2012/12/coord-decoder.html>