

This seems to be a typical signal from the HX711:

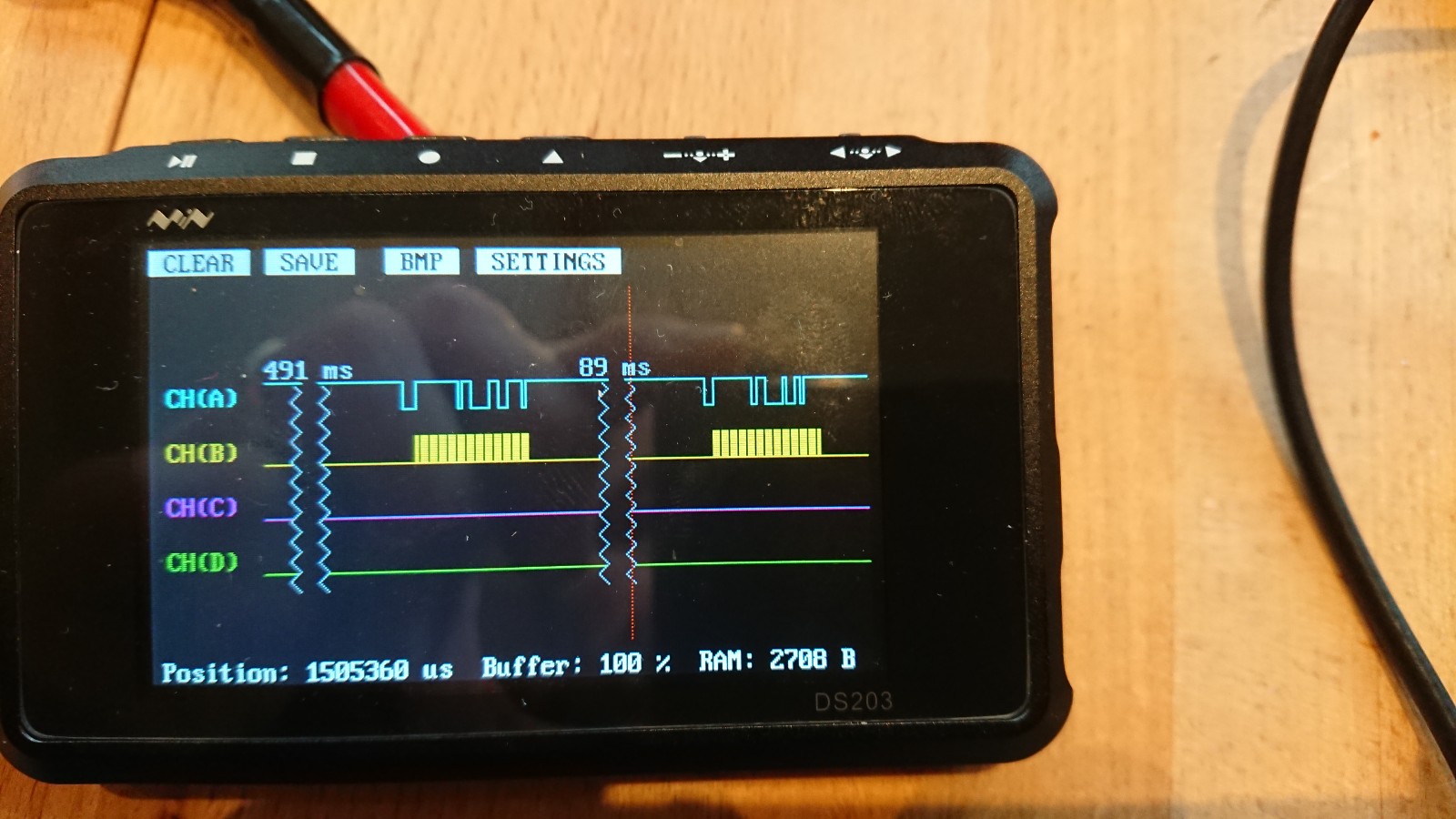
DOUT (blue) goes low, saying data is ready, then PD SCK is pulsed to see the data at DOUT

OK, works nicely, made some stupid errors: 1) swapped DOUT and SCK, nevermind

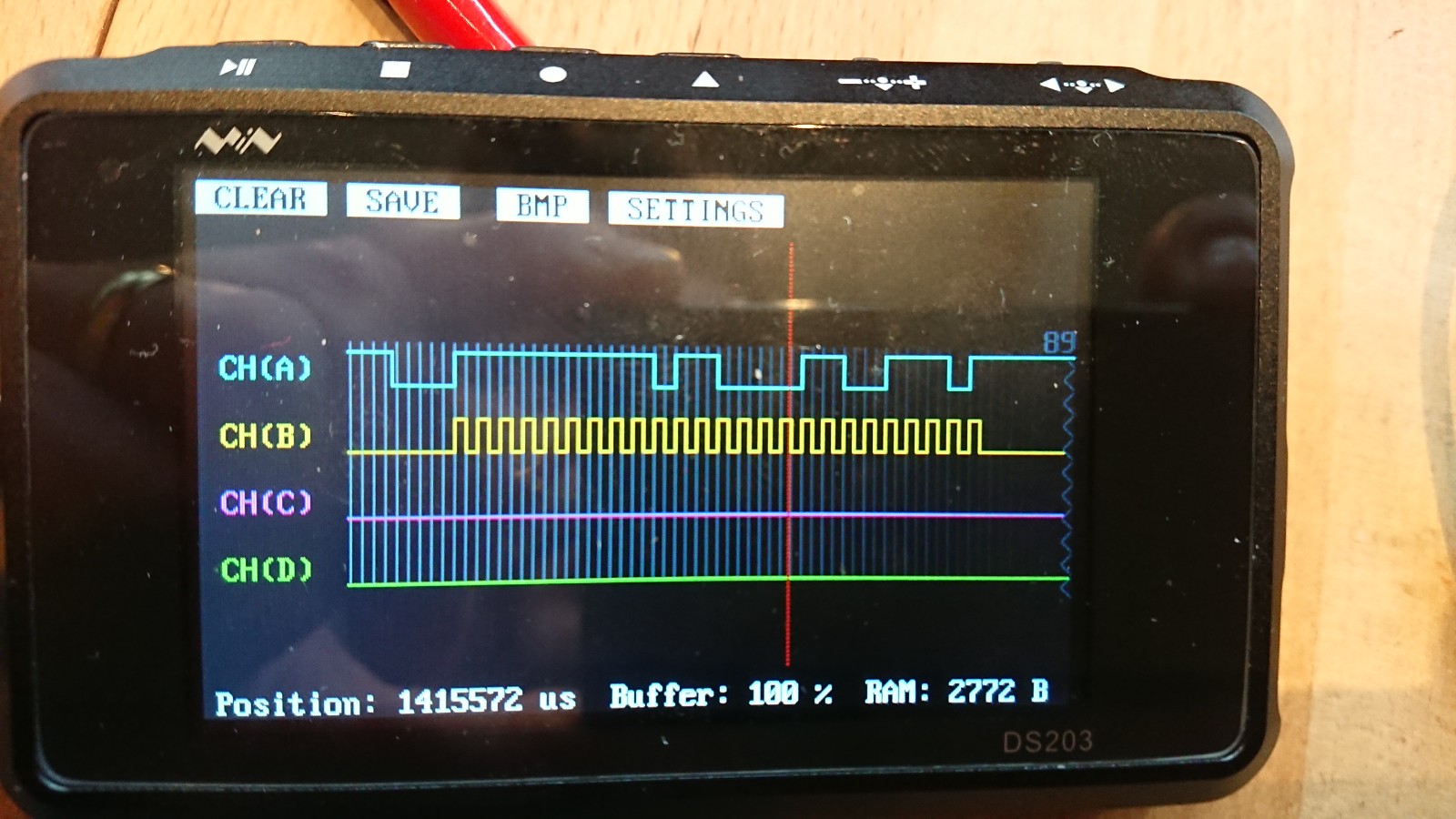
2) installed the RMT TX channel using rmt\_driver\_install

Which is at odds with having your own interrupt.

For debugging purposes I have reduced the RMT pulse unit to 1 us and use pulses of length 10 ie 10us

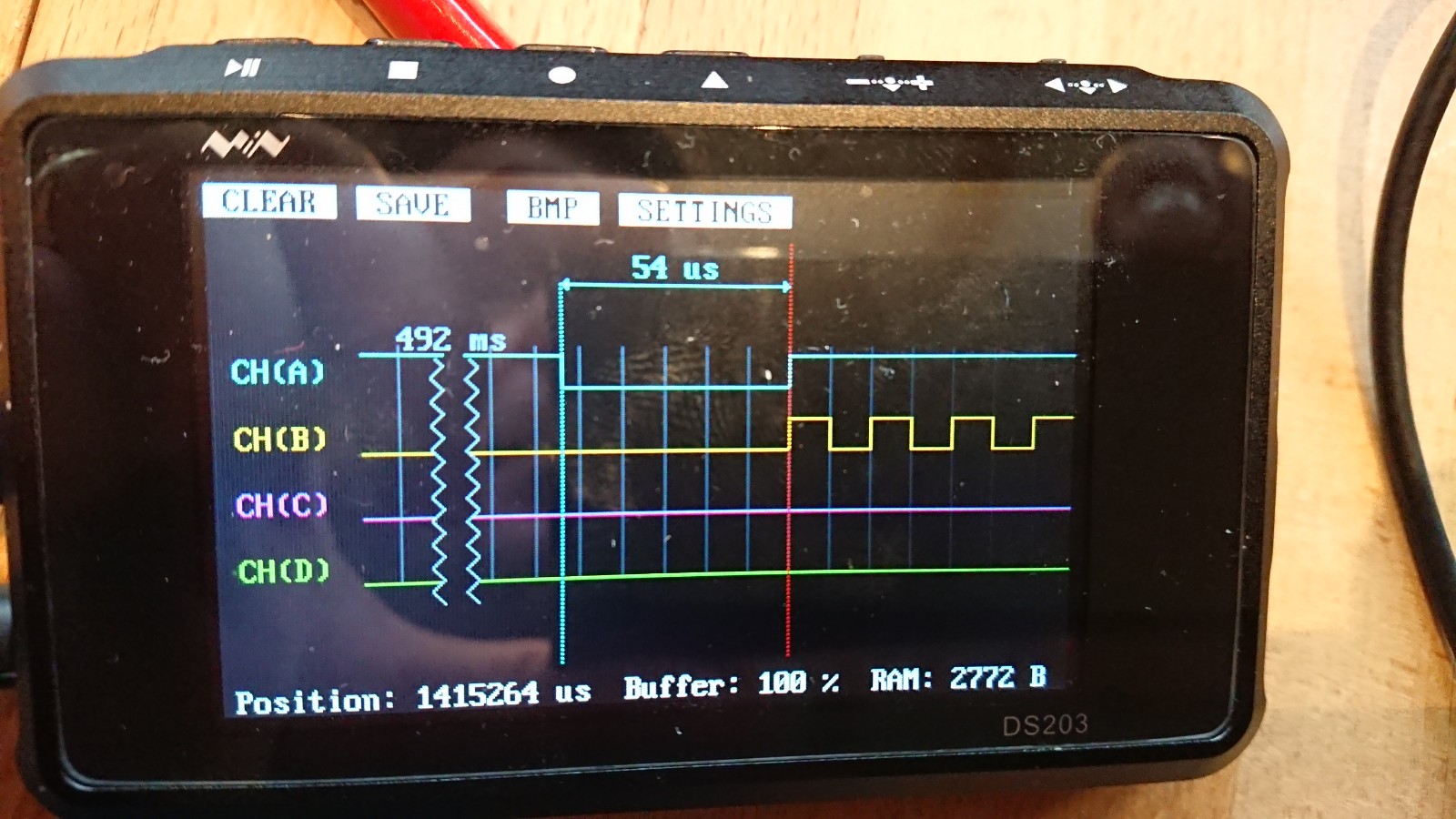


Nice reading every 90ms equals SPS of about 10



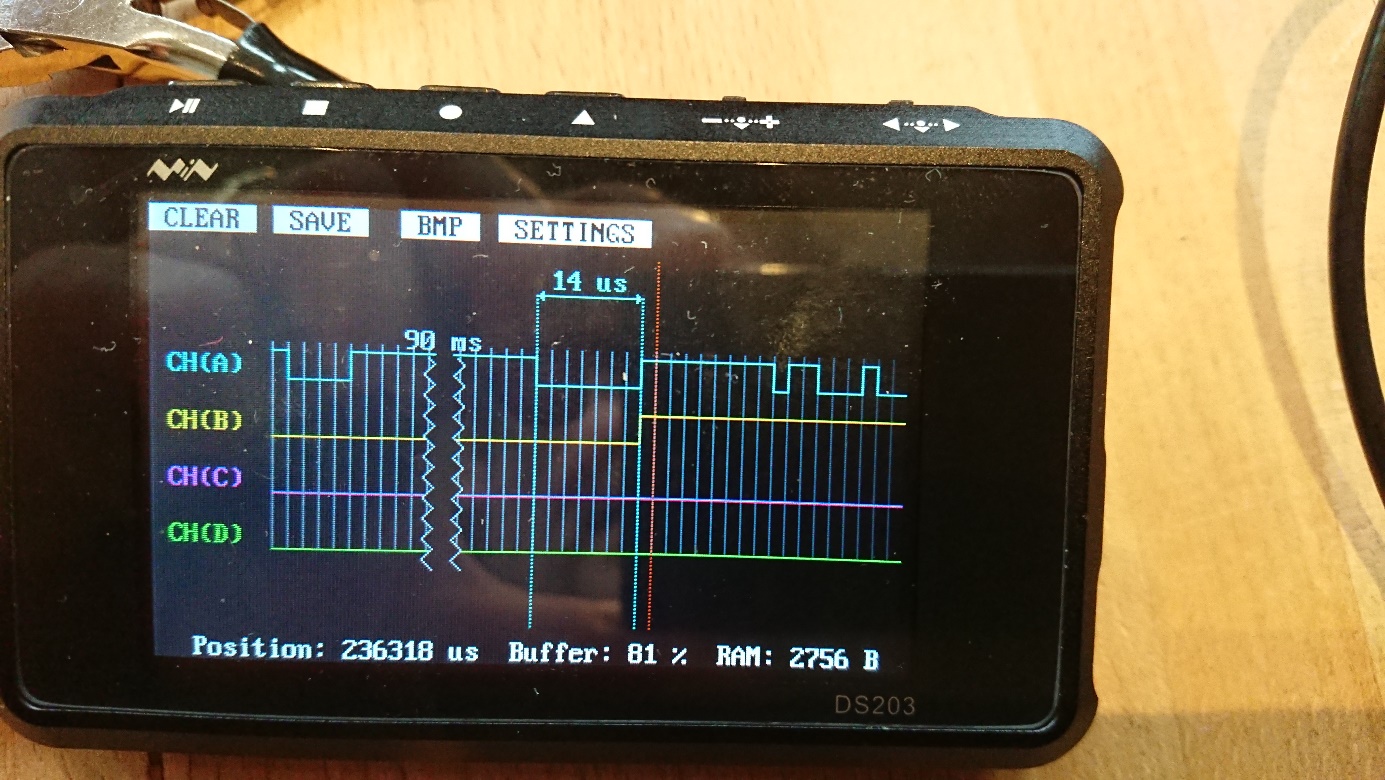
Nice strobe signal (yellow) and nice response (blue) on DOUT

Latency between falling edge of DOUT (data ready) is about 54 us



BUT this is using the slow timing for debugging. Let’s see how fast we can go:

Latency is now 14us (note that the scope does not resolve the SCK pulses anymore)

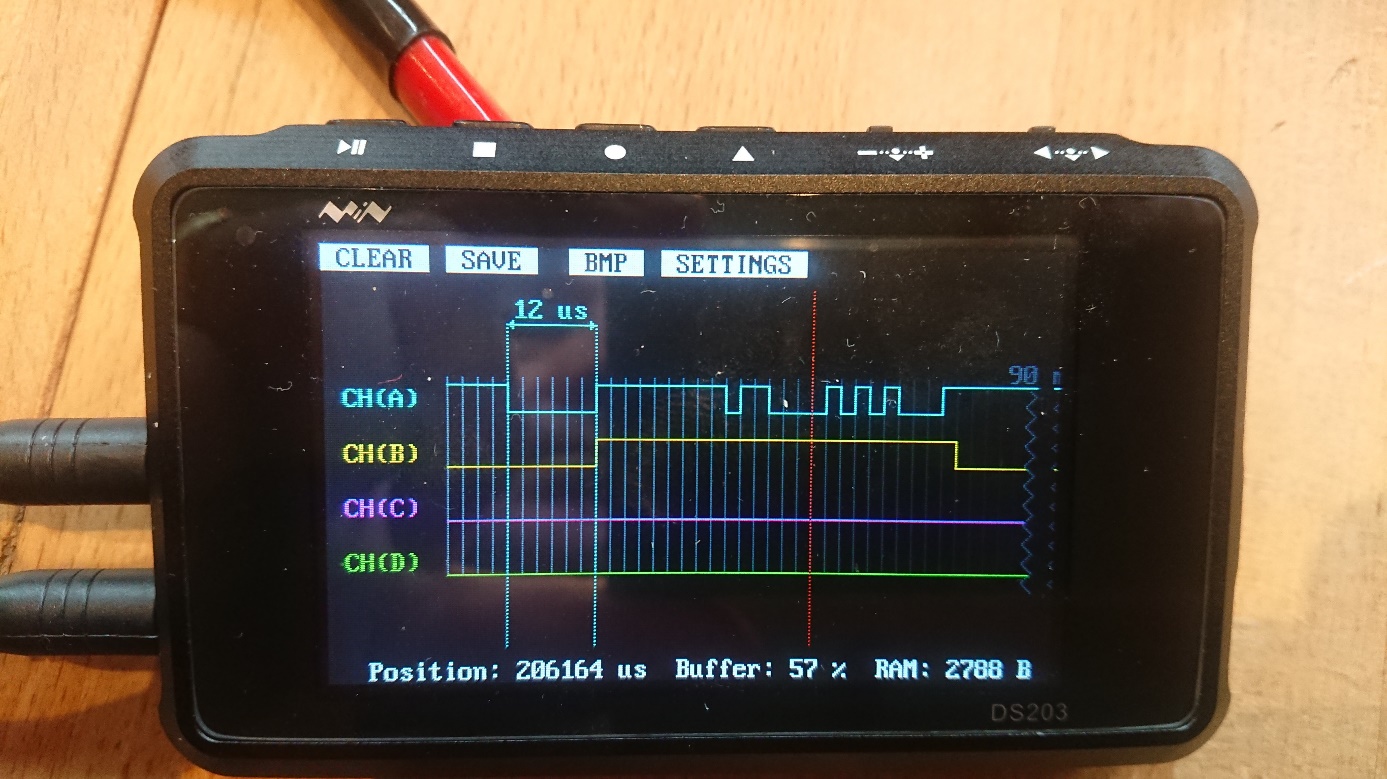


If we get rid of the first 2us „delay pulse“:

    {{{10, 0, 10, 0}}},     // keep L for 2us (because of ISR latency prob not needed)

    {{{10, 1, 10, 0}}},     // H for 1us and L for 1us (line is pulled down)

It looks like this, so effectively we have a 12us delay between falling edge of DOUT and rising edge of first SCK strobe



Now we need to decode the pulses. First idea:

Define 24 midpoints in strobe train using the 12us delay then loop through pulses from RMT,

            b1 = (data[i].duration0);

            b2 = (data[i].duration1);

data[0].duration0 is the duration of the first negative pulse

looking at pulse durations for the first (negative pulse) we see:

pulses = [164 264 284 304 324 344 364 403], so nicely on a 2us raster (20 counts = 2us)

so first negative pulse seems to have a minimum at 16.4us