

## Setup: Espruino Pico and SH1106

### Code

```
SPI1.setup({mosi:B5, sck:B3, baud: 100000}); // varied this between runs

g = require("https://github.com/jugglingcats/EspruinoDocs/blob/master/devices/SH1106.js")
  .connectSPI(SPI1, B7, B6, function() {

  var c=0;

  function draw() {
    if ( c++ % 2 === 0 ) {
      g.fillRect(0, 0, 127, 63);
    } else {
      g.clear();
    }
    g.flip();

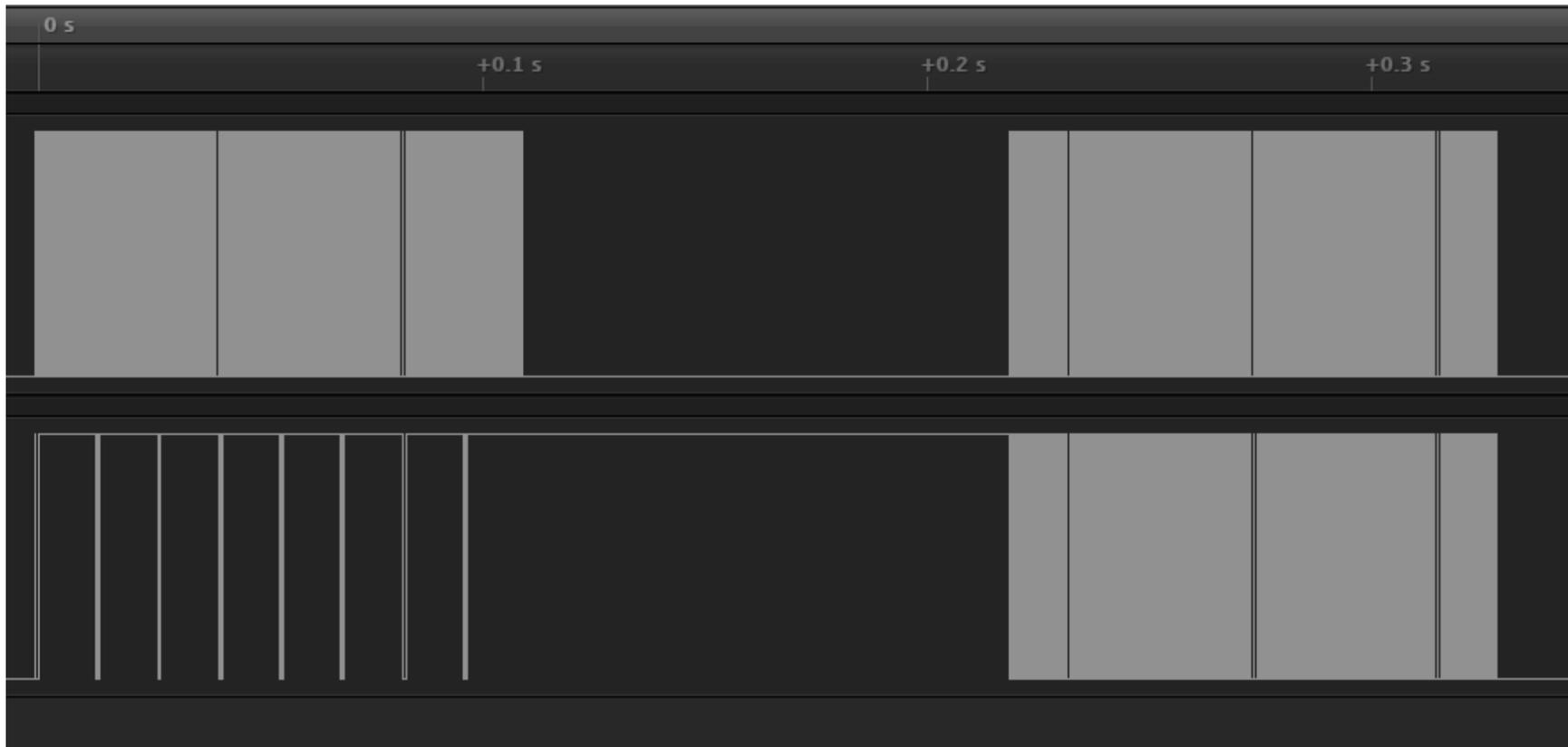
    setTimeout(draw, 100); // varied this between runs
  }
  setTimeout(draw, 0);
}, {cs: B15});
```

The code alternates between painting an empty and full screen, so the data being sent is either all 0's or all 1's.

All images show CLK on top and MOSI on bottom.

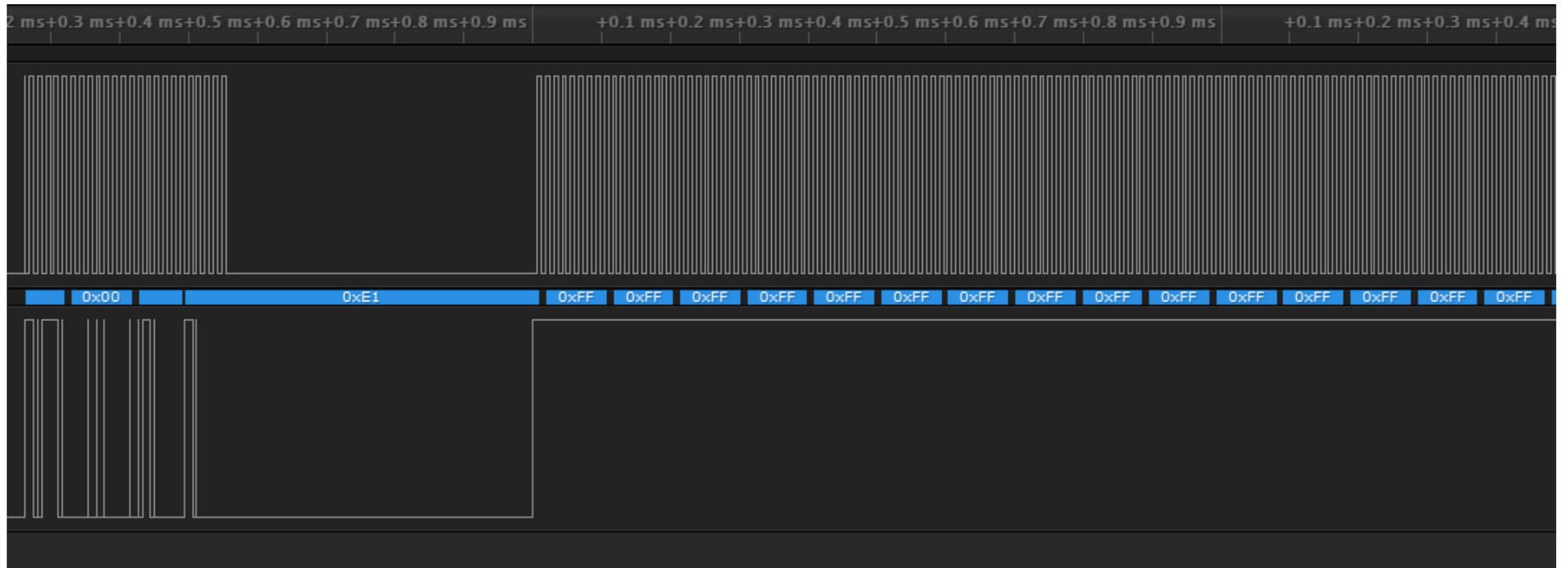
TLDR; skip to the last page for conclusion

For f baud at default 100,000. Repainting the screen is quite slow – as you can see below each paint takes about 100ms – 10 fps maximum.



The big gaps are due to the 100ms delay in setTimout, just so we get clear separation in the data.

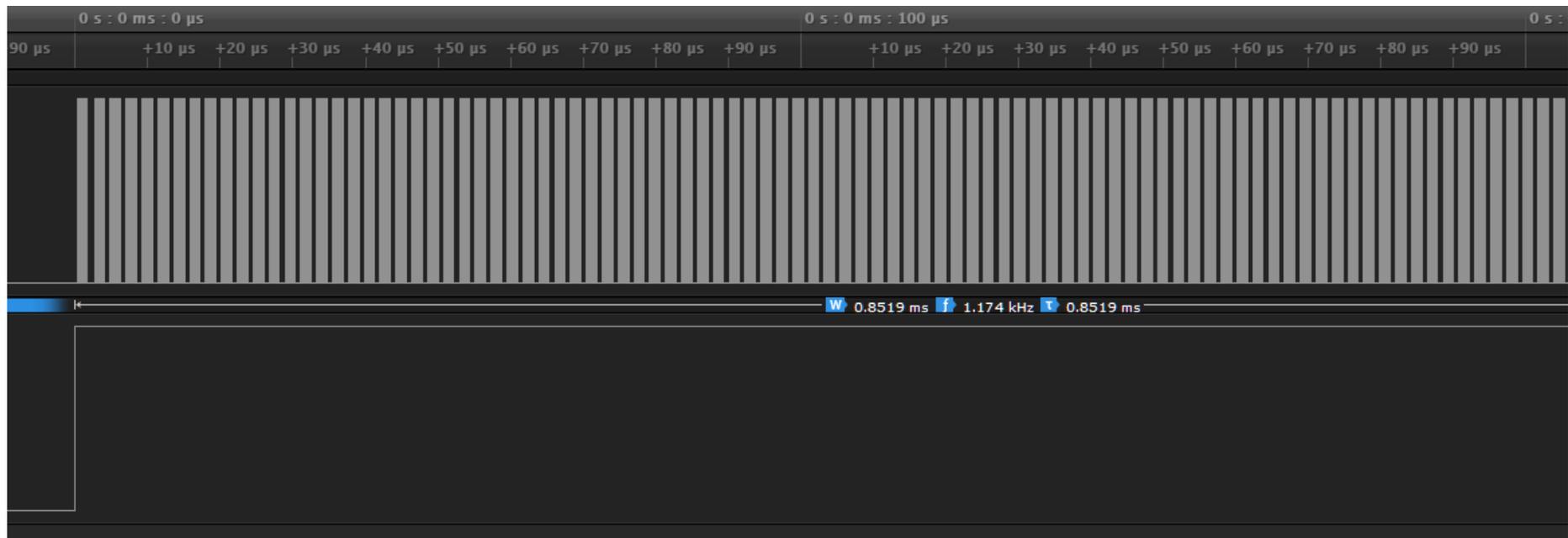
Zooming in you can see there are no real gaps in the data. We see some setup and then nice clean strings of 0xFF.



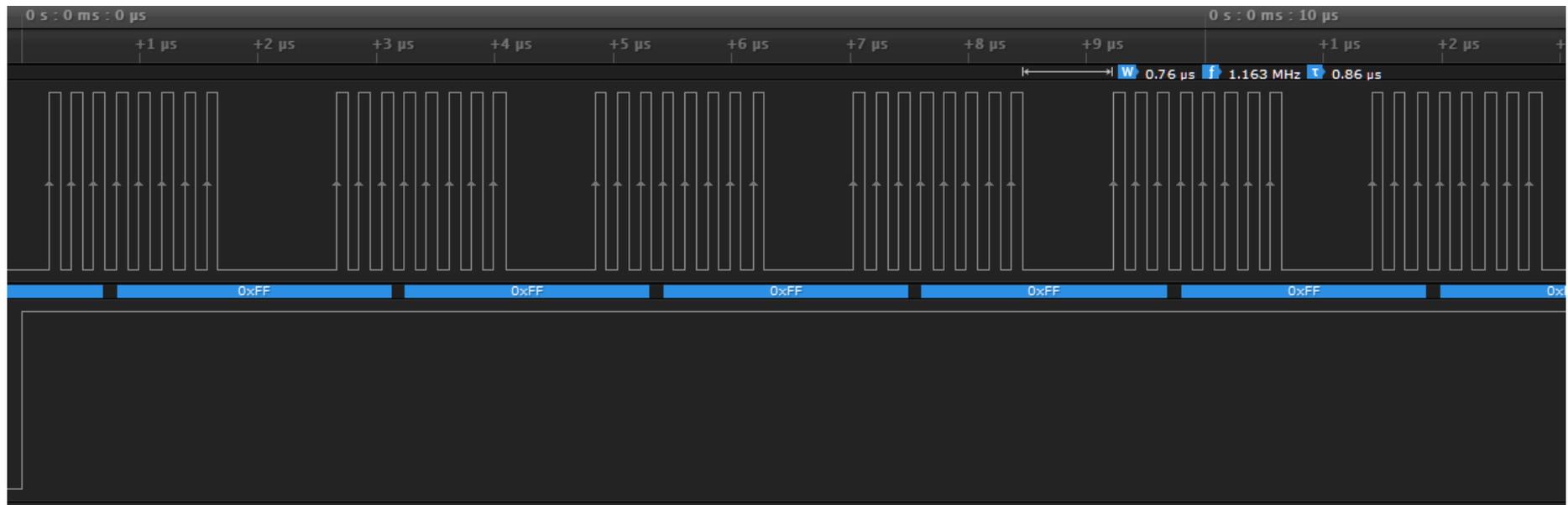
Increasing the baud rate to 4000000 and things start looking less tidy. Here is macro view of a single “screen full” of data:



It appears that less than 50% of the SPI bandwidth is utilised. It is actually worse. If we zoom in once on an area of clock activity:



It looks like good solid activity. But if we zoom some more:



We can see that between each byte send there is a pause.

**But really we don't need this frequency...!**

A happy setting is about 800,000 baud where each repaint takes around 20ms... Here I set the timeout interval to 25ms.

