I2C Clock Stretch Probe

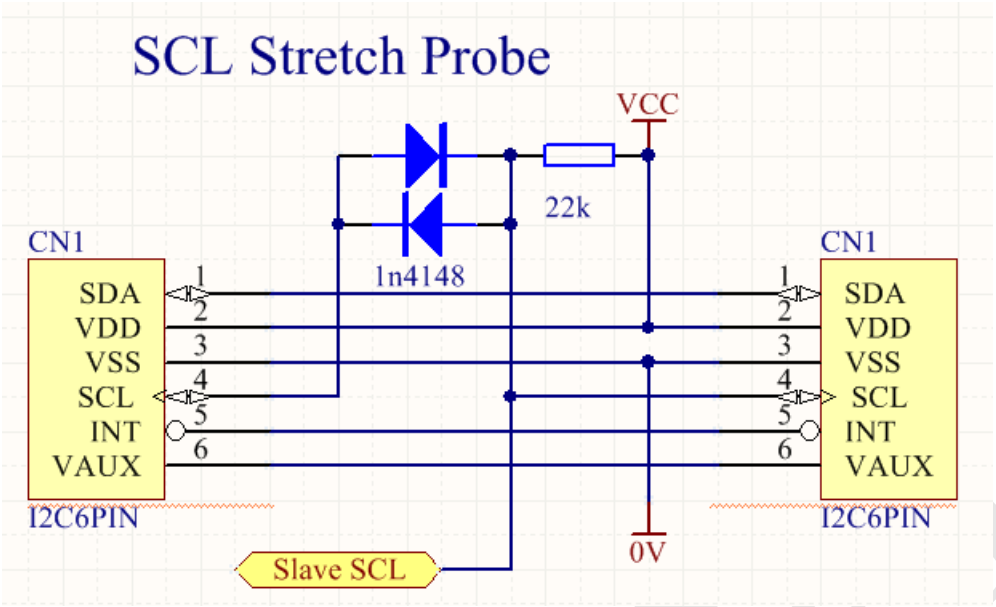
1 Introduction

I2C allows a clock synchonising technique called Clock Stretching, where the slave can hold the SCL line low while it is busy. Clock Stretching is not easily seen when viewing the SCL line with an oscilliscope - this show you how to make a very simple probe to view and trigger off it. The I2C specification puts few limits on clock stretch, and this means that is can cause unpredictability in timing and hard to find intermittant errors, which makes viewing it, and triggering off it, important.

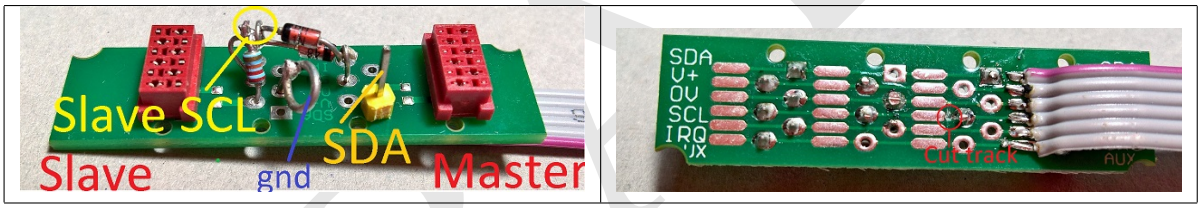
Contents

2 Simple Probe

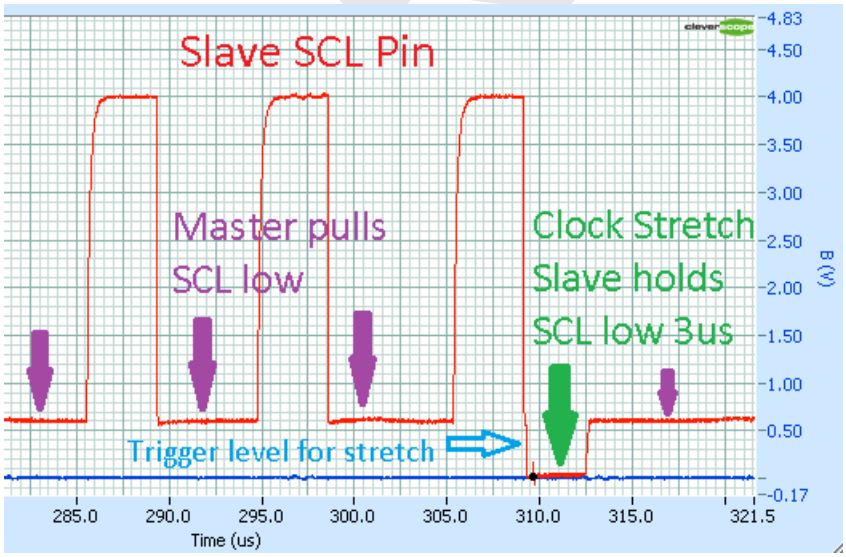
A very simple circuit will makes it easy to see the stretching.



This is easily made on the small hub pcb that comes with the I2C Starter Kit or connector kits. Any diode will do, signal or power diodes.

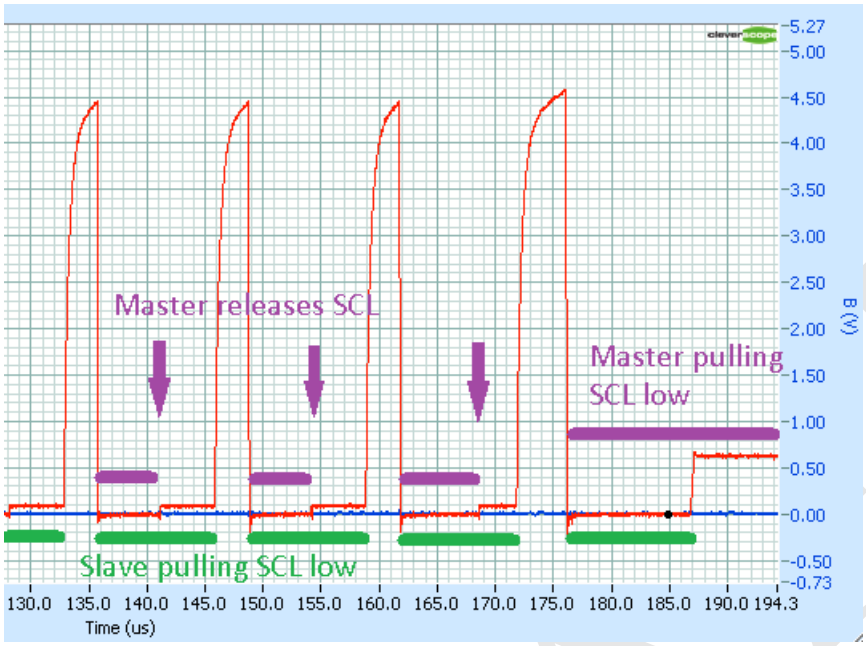


For 5V systems use ordinary silicon diodes, any diode will do. For low voltage I2C, or where any kind of bus isolator is used, schottky diodes might be better



Attach the probe to the Slave side SCL. Here we see a Microchip PIC based slave stretching the clock for 3us after the ACK clock pulse. Note that the clock is not *actually* being stretched, as the master is still holding the bus low when the slave releases SCL.

To trigger off clock stretching points just set trigger level to 0.4V. If you are using an advanced scope like cleverscope you can set time and level trigger, to find only clock stretches longer than a chosen time.



Here we see a slave (with weak outputs or series resistance), which clock stretches every clock pulse. In this case it is stretching SCL within the first 8 bits - the master releases SCL before the slave releases it. The small step in level makes it clear where the the master has released

3 Clock Stretching

Dedicated hardware I2C chips seldom use it, but slaves based on microprocessors, are almost obliged to use it as they have to at least respond to an interrupt to prepare data when a read is requested. If your system is using a processor based slave or an SOC type slave, it will often

The I2C specification puts few limits on clock stretch, and this means that is can cause unpredictability in timing and hard to find intermittant errors.

- It can be an unlimited length in I2C, thus stalling the bus or causing timeouts (e.g. our BL233B device times out at 17ms)
- It can be after every bit, or just once per byte
- Different chips do it differently e.g. PIC stretches after ACK, PSOC1 stretches before ACK.
- long stretches will extend out to hit Start and Stop operations

3.1 Bit-bash I2C Masters

These have to be very carefully coded[1] to reliably support all stretch possibilities and meet timing specifications, if you are using a library (eg Arduino) it likely doesn't work properly.

[1] yup, we have found a bug in our own implementations

Not only the byte operations, but also start and stop have to correctly handle stretch.

3.2 SMBUS

The SMBUS variant does impose a maximum time on stretch. I2C slaves could exceed this and cause bus errors.

4 Revision History

Rev	Date	Changes
0		First Release
1		
2		
3		