

## Designing glitch free opto- isolated I<sup>2</sup>C- bus systems

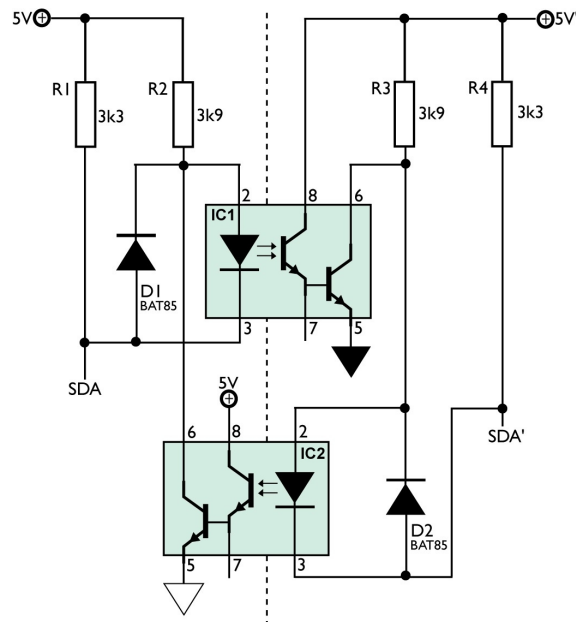


Figure 1. National Semiconductor Application circuit ca.1998.

Almost every year a 'new' design idea appears in the technical press claiming to achieve opto-isolation of an I<sup>2</sup>C- bus. They mostly 're-discover' the arrangement shown in Fig.1, a circuit from a National Semiconductor Application from many years ago, which is typical of the published circuits for opto-isolating the I<sup>2</sup>C- bus. It probably was not the first, and certainly not the last!

Because opto-isolators can only handle unidirectional signals it is necessary to split the bi-directional I<sup>2</sup>C- bus signals into two uni-directional signals. The way to do this without causing 'glitches' on the bus logic signals is to use a buffer based on the principle of P82B96/ PCA9600 that splits the bus by using special logic levels on one side to avoid the glitches.

Most other attempts, such as the one illustrated in Fig. 1, are based on the concept of temporarily blocking the bi-directional I<sup>2</sup>C signal propagation in one direction in order to prevent the bus latch-up caused by feedback of the logic signals [TR002: Techniques for buffering I<sup>2</sup>C signals.](#)

As a result these all produce a bus 'glitch' during the propagation time of the logic control signals that must reverse the blocking process when the bus driving signals change. All these early attempts resulted in, and most admitted to, problems with glitches resulting from those signal propagation delays. Some would even oscillate for certain values of bus loading capacitance.

This arrangement provides a useful example to illustrate the problems with all these time-based 'solutions'. To check whether an isolating circuit will generate unwanted glitches on the bus, there is a very simple test that can be applied.

### **Apply this test to reveal whether an I<sup>2</sup>C arrangement will produce problem glitches:**

- Drive the I<sup>2</sup>C- bus, connected to one side of the circuit arrangement, LOW.
- Then, holding that first side LOW, drive the I<sup>2</sup>C- bus connected to the other side LOW.
- Release the bus drive on the first side.

- If the system conforms to the I<sup>2</sup>C protocol the first side should just stay LOW, held LOW by the second side.

The test can be applied in either of these two ways – simply follow the sequence with an oscilloscope connected to the side that is driven LOW first, and then released, and watch for the glitch when it is released – or mentally work through the sequence of events as illustrated in this following example.

**Apply this ‘glitch’ test to the circuit of Figure 1. Follow the sequence of events to show what happens.**

- Drive the left side SDA LOW. Current in R2 turns on the LED in opto-coupler IC1. After a short delay the photo-transistor in IC1 turns on and the isolated SDA' is pulled LOW via diode D2.
- Now externally drive the I<sup>2</sup>C- bus connected to the right hand side SDA' LOW. Remember it is already LOW so the external LOW at SDA' does not change the state of that I<sup>2</sup>C- bus. That is normal. The photo-transistor in IC1 remains on, so there is no current in the LED of IC2. The photo-transistor in IC2 isn't on either.
- Now release the I<sup>2</sup>C- bus at SDA. SDA goes HIGH because there is nothing to hold it LOW. The photo-transistor in IC2 isn't on so SDA is pulled HIGH by R1 (and initially with some contribution by R2).
- SDA rises with a rise time determined by the bus capacitance loading at SDA and the pull-up resistor R1. (During the initial part of its rise there is also some contribution by R2 via the LED in IC1.) After SDA rises above 30% of the supply\* (30% of 5 V) the I<sup>2</sup>C- bus at SDA is no longer guaranteed to be at the bus LOW level. This represents the start of an unwanted ‘glitch’ of the I<sup>2</sup>C- bus at SDA. It should be LOW, because SDA' is LOW, but now it's HIGH.
- When SDA rises high enough, the LED of opto-coupler IC1 turns off. After some delay time its photo-transistor will turn off. After the photo-transistor in IC1 turns off, R3 can source current to the LED in opto-coupler IC2. That current flows, via the LED, to ground via SDA' which is externally being held LOW. After another switch-on delay the photo-transistor in IC2 turns on and SDA is pulled down again to the correct LOW state. That represents the end of the unwanted ‘glitch’ to the HIGH level on the I<sup>2</sup>C- bus at SDA.

To determine the duration of the unwanted glitch, notice that SDA is HIGH during the time taken for IC2 to turn off plus the time taken for IC1 to turn on. Those two propagation delay times should be found in the opto-isolator's datasheet.

The false bus HIGH signal can lead to serious problems. On the SCL bus line it would represent an additional false clock pulse, causing all following data bits to be incorrectly read, and the Acknowledge to be provided at the wrong time.

One of the less obvious problems is that there may also be a significant, but poorly defined, delay introduced by a slow rise of the bus at SDA after the external drive signal was released. The I<sup>2</sup>C is not guaranteed to be LOW if the voltage is above 30%Vcc (in this case 1.5 V) but the LED current in IC1 does not fall to zero until the voltage at SDA rises to within about 1.5 V (the LED on-voltage) of the 5 V rail. Even assuming a typical logic switching level of Vcc/2 for the I<sup>2</sup>C- bus at SDA, that bus can be HIGH (above 2.5 V) for some time before the LED in IC1 turns off. The ‘glitch’ time can be quite long.

\* *Note:* For simplicity, and because these circuits appeared before Fast-mode was widely used, this document uses the easily recognised Standard-mode bus V<sub>IL</sub> value of 30%Vcc. The Fast-mode specification mandates hysteresis that changes the bus level requirements. For example the bus LOW requirement, including hysteresis and noise margin, then becomes 0.25Vcc. The bus LOW requirement for Standard-mode, including noise margin, is actually 0.2Vcc so any glitches above these 0.2Vcc or 0.25Vcc may cause non-compliance with the specifications. For more details see [TR004: Noise Margin in I<sup>2</sup>C signals](#).

---

***These circuits obviously work 'sometimes', so when CAN they be safely applied?***

There is always some risk in allowing false logic 'glitches' on the bus lines because their effect on the many different devices that may be connected to the bus can't ever be known with certainty.

Supporters of the approach will claim that the recommended test above applies to the special situation known as 'bus contention' that happens only in multi-master I<sup>2</sup>C systems as masters negotiate for control of the bus and that it is no problem in single master systems. They may also claim that another situation that results in this contention is 'bus stretching' and that slaves that have bus stretch capability are either very rare or maybe no longer even available.

Both ignore the most common occurrence of bus contention and are therefore dangerous conclusions.

They do not take account of the normal bus Acknowledge signal that happens during every initial bus communication, even in a single master system and with any type of slave.

Consider the case of a single master writing the initial address word to a simple single slave. The 8<sup>th</sup> address bit written to the SDA bus by the master may be a HIGH or a LOW but will more often be a 'write' bit, a LOW. The next data bit to appear on SDA, to be provided by the slave, is the Acknowledge bit. It is also a LOW.

There is usually no guarantee about the relative timing of the end of that last LOW (write) bit coming from the master and the Acknowledge LOW provided by the slave. Both happen some time after the falling edge of SCL and during the LOW period of SCL, but their relative timing is rarely guaranteed. If the master's write bit happened to be removed late in the SCL LOW period it is entirely possible that the Acknowledge

LOW from the slave will be disguised by the false glitch to HIGH produced by these signal blocking circuits. If that false glitch to HIGH extends into the required data set-up time then bus errors are possible. When the glitch times are relatively long, or poorly defined, they create a real potential for error. Selecting a slow clock speed, checking for published t<sub>VD</sub> times for the I<sup>2</sup>C parts, and carefully calculating the system timing can make it reasonably safe to use this technique on the SDA line

One thing is clear, this approach should never be used on the SCL bus signals because the spurious clock pulses produced are almost guaranteed to result in system errors.

In single master systems, in cases where clock stretching will not happen, the SCL signal may be made uni-directional. The arrangement of Fig.1 will then not be used for the SCL line, a simple open collector/drain buffer is used instead. That SCL signal may then even be passed through an opto-isolator or have additional delays. In that case, and after carefully checking the system timings, reliable systems based on this approach for handling the bi-directional SDA signal certainly can be built.

***Conclusion:***

Only the P82B96/ PCA9600 approach guarantees a reliable glitch-free operation in applications that require the I<sup>2</sup>C signals to be split into uni-directional signals to pass via opto-isolators. That arrangement also retains full multi-master system capability.

When the system only has a single master, and a uni-directional SCL line can be used, then lower cost arrangements can provide acceptable alternatives. They will generate some glitches on the SDA line but those glitches can be designed to occur only at times that are not important for correct operation.

Designing an I<sup>2</sup>C system? Visit [www.bus-buffer.com](http://www.bus-buffer.com) for more information or to contact us.