Errata Sheet

ES000131

PCAP04

Errata Document of Datasheet Version 3-00 or later

One issue has been identified which can affect performance and is described in this Errata Sheet.

Deviation 1: I²C Multi-Slave Operation

Description:

PCap04 is responding and acknowledging its entire opcodes even if the I²C address is not targeting that device. For instance, a write opcode with a false I²C address still lets the chip to get the received data into its memory. This bug causes the following problems:

- No multi slave communication possible
- After start condition, PCap04 acknowledges any byte that resembles to its opcodes on every I²C address
- Chip registers / SRAM can be changed unintentionally

Counter measures can be:

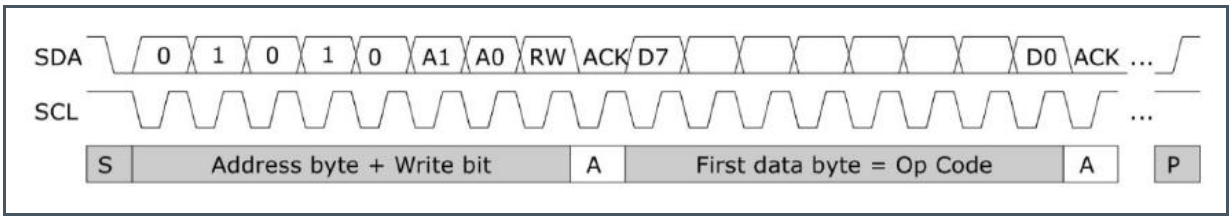
- IIC_EN input of PCap04 can be used as chip enable as workaround (SSN_PG0 := DVDD; configure as input; MISO_PG1 := DVDD; configure as input)
- I²C address devices like PCA9540B can be used as long as no valid PCap04 opcode is used

Detailed Description:

The master starts communication via I²C interface by sending the start condition and then the I²C address of the device on the SDA. However, the PCap04 acknowledges this data-packet when it complies with one of the PCAP04 opcode, even though the address is not matching and it is not acknowledged.

The communication with the PCap04 (slave) through I²C interface is done as shown in Figure 1:

Figure 1:
I²C Typical Sequence

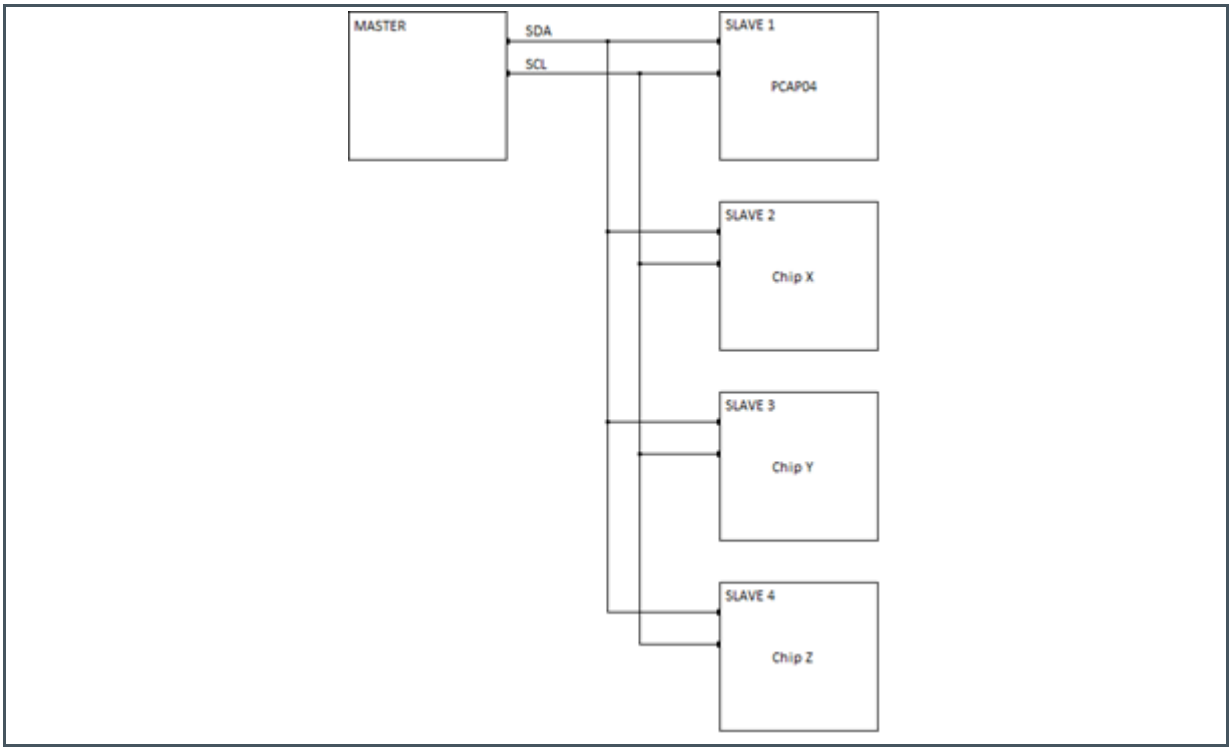


After the start condition, the correct I²C address has to be sent by the master and the PCAP04 has to acknowledge it to continue the communication. The I²C address is a 2-bit value located in register 0, bits 7:6.

In a bug-free multi-slave interface, all the rest of the data delivered by the master must be rejected (not acknowledged) by the slave in case the I²C address does not match the chip address.

This is important especially if the master monitors the data line and verifies constantly its functionality. In such a system, only one slave should acknowledge when the master sends a data (opcode, address etc.). If the master observes more than one acknowledge at a time and if it is a smart interface then it may stop the communication.

Figure 2:
Multi-Slave I²C Interface Circuit.



Due to the bug in PCAP04, this “more than one acknowledge” situation can happen. Suppose that in Figure 2 the master communicates with chip Y. Therefore, it first creates the start condition and writes

the I²C Address of chip Y into the SDA line. Then the address has to be acknowledged by chip Y. In parallel, since PCAP04 cannot identify the address it simply rejects it (not-acknowledge), so as in chip X and Z. Everything is ok so far. After that however, if any data sent by master to the Chip Y resembles to any opcode of the PCAP04, then PCAP04 would acknowledge it too and drive the interface into 2 acknowledge at a time situation.

Workaround:

The following workarounds that can be taken into consideration if PCAP04 needs to work in a multi-slave system.

1. Use the IIC_EN pin as a chip select

This workaround for the bug involves an extra wire in the I²C connection. SSN_PG0 and MISO_PG1 to be tied to DVDD and configured as input.

PCAP04 is capable to communicate with peripherals either via I²C or SPI. The interface is selected by the digital input of the IIC_EN pin. If the IIC_EN pin is HIGH then I²C is active, if it is LOW then SPI is active. In this workaround, this pin is used as a chip select.

Figure 3:
Example Circuit with an Extra Chip Select Line

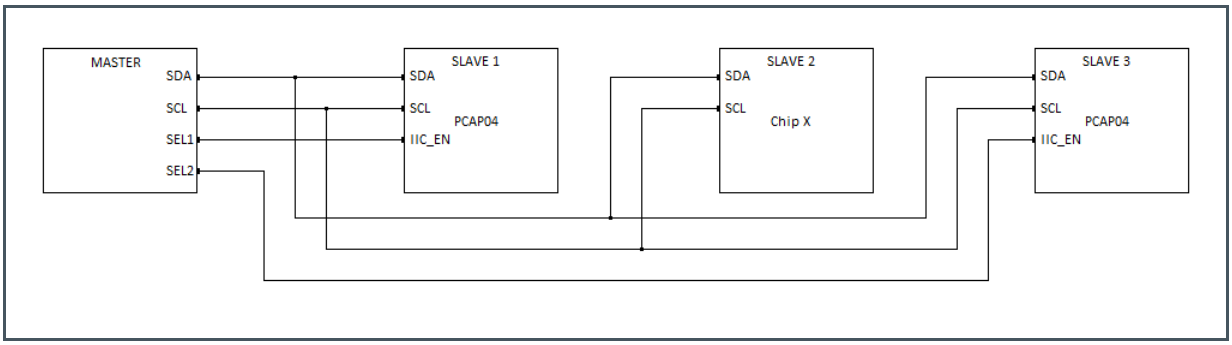


Figure 3 shows an example. If the master needs to communicate with slave 1, then SEL1 signal has to be high. This will activate the I²C interface in the Slave 1 (PCAP04) and allows the common I²C communication between itself and the master. During this communication, SEL2 signal must be low, of course, which deactivates the communication in Slave 3.

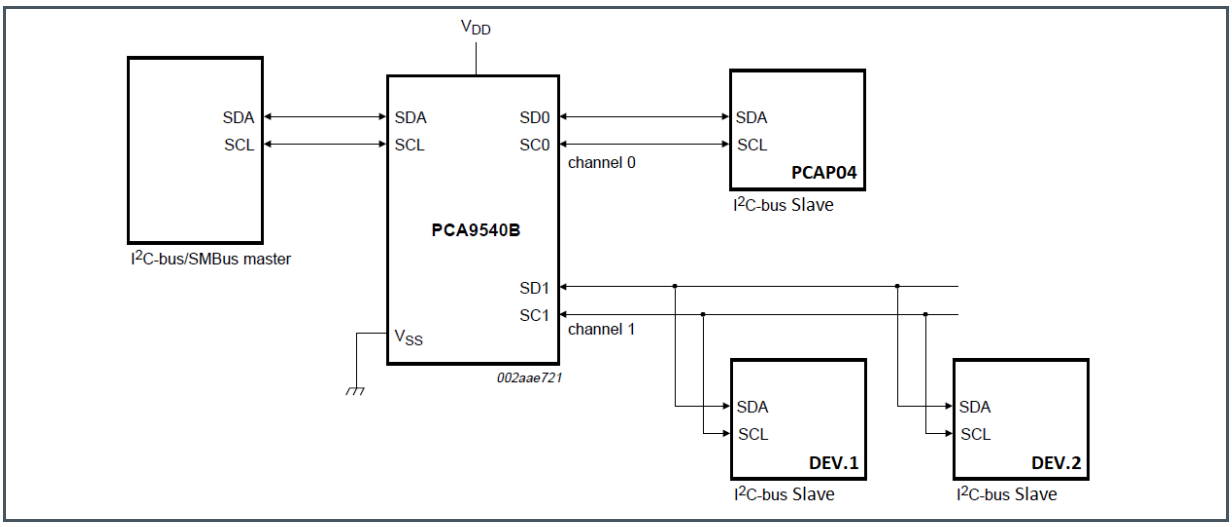
This workaround was tested and verified. It is the easiest workaround for the bug, but needs an additional line.

2. Using a Bus Multiplexer

Another option for the use of PCAP04 in a multi-slave system is an additional I²C bus multiplexer like e.g. PCA9540B from NXP. It is a bidirectional two-channel multiplexer. It has a control register which is used to hold the target channel of the communication.

It is supposed that the devices other than PCap04 are suitable for multi slave interface. Therefore, the multiplexer is only used to isolate the PCap04. Figure 4 shows an example, in which channel 1 is used to address all other devices besides PCAP04.

Figure 4:
Using a Multiplexer



3. Using GPIOs to listen to the interface and control IIC_EN pin as a chip select

Figure 5:
Using GPIOs to Verify the Interface and Control IIC_EN Pin.

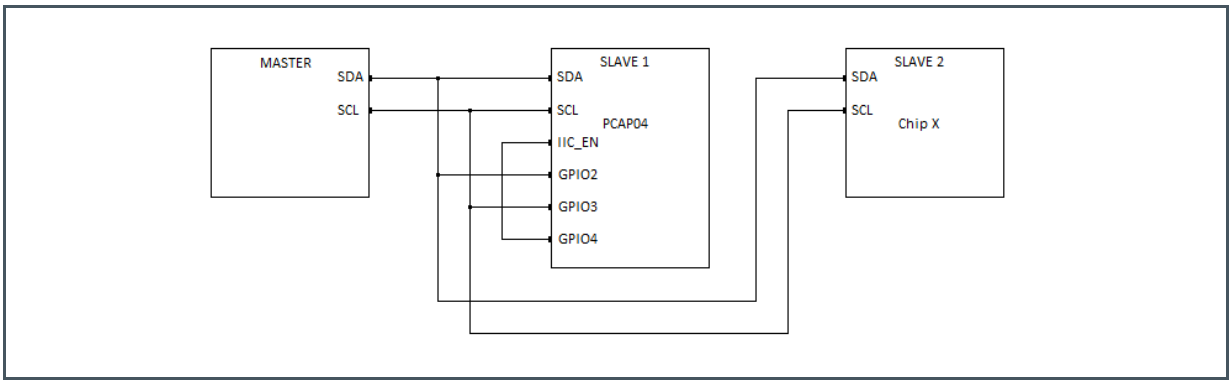


Figure 5 shows the circuit diagram of this workaround. The idea is, again, to use the IIC_EN pin as a chip select that activates the I²C interface only when needed. Please read first the workaround 1 to get enough insight into this workaround. However, instead of using a third wire just like in workaround 1, the control of IIC_EN pin is done in a way that is more sophisticated.

Here the GPIO2 and GPIO3 are assigned as inputs and GPIO4 is output. By using a dedicated firmware, the SCL and SDA pins should be listened constantly by the two GPIOs. The firmware has to check the first 8 bytes following the I²C start condition. If the address is matching with the chip's address, than it sets e.g. GPIO4 to 1, and as this is connected to IIC_EN it activates the I²C interface.

After that, however, the Master has to restart the interface and resend the chip address to be able to communicate with the chip.

The very first communication needs some action regarding the acknowledge. Since the GPIO2 port is suitable for input and output directions, an acknowledge may be coded in the FW. Another action could be to set the master to send the request two times, and to ignore the first not-acknowledge situation. A full verification of the approach is outstanding. And it costs area in the NVRAM (Firmware), three GPIO ports and more current to keep the CPU always running.

Affected Devices

Figure 6:
Affected Devices

Manufacturer Part Number (MPN)	Product Description	Material ID
PCAP04-ASDF	PCAP04-ASDF SD F	502040012
PCAP04-ASWB	PCAP04-ASWB SW	502040013
PCAP04-BQFM-24	PCAP04-BQFM-24 QFN24 LF T&RDP	502040021
PCAP04-BQFT-24	PCAP04-BQFT-24 QFN24 LF T&RDP	502040020
PCAP04-AQFM-24	PCAP04-AQFM-24 QFN24 LF T&RDP	502040014
PCAP04-AQFT-24	PCAP04-AQFT-24 QFN24 LF T&RDP	502040010
PCAP04-EVA-BOARD	PCap04-EVA-BOARD	220300002
PCAP04-EVA-KIT	PCap04-EVA-KIT V1.0	220300003

Summary

Due to this bug in the I^2C interface the PCAP04 the chip is unfortunately not suitable to connect directly to a multi slave system. This may arise “two-acknowledges at a time” situation and the master device may interrupt the communication due to this. Such situations can be avoided by using a I^2C bus multiplexer. Alternatively the IIC_EN pin can be used to enable/disable the I^2C interface.

Headquarters
ams AG
Tobelbader Strasse 30
8141 Premstaetten
Austria, Europe
Tel: +43 (0) 3136 500 0