Analysis of sensor disturbances caused by IEMI

Arne Pahl, Faculty of Electrical Engineering, Helmut Schmidt University, Hamburg, Germany
Prof. Dr.-Ing. Stefan Dickmann, Faculty of Electrical Engineering, Helmut Schmidt University, Hamburg, Germany

1 Introduction

Sensors are the link between the digital and the analog world. They can be found in almost all electronic systems and provide the system with access to environmental data. The disturbance of such enviromental data can have serious consequences. For example, an oil pressure sensor on a U.S. Navy ship was disturbed by an RF signal in such a way that a shutdown signal in the automatic power control system was triggerd. Likewise, some early ABS systems in Germany were affected by a radio transmitter in such a way that there were severe problems with the brakes in certain highway sections [3]. Although sensors systems are susceptible to interference, there is insufficient research on the susceptibility of sensors systems to interference. There are studies on the susceptibility of microelectromechanical systems (MEMS) to acoustic signals [7, 8]. However, it was pointed out in [5] that electromagnetic interference is generally not explicitly addressed or even neglected in sensor design.
In our previous work [4] we have shown that a MEMS sensor with Inter-Integrated Circuit (I²C) interface can be disturbed by electrical fields. The effects include communication breakdowns, controller crashes, short term and permanent sensor value errors. In [2], similar errors are reported for two other sensors with I²C interface.
In this work, we focus on the cause of the observed interference by localizing the interference susceptibility and analyzing the I²C communication. First, the measurement setup and the sensors used are presented in Section 2. The measurement results are shown in Section 3. Finally, in Section 4 we give a short conclusion and an outlook.

2 Measuring Principle

In [4] and [2], the disturbances described were caused by an electric field. In order to investigate these disturbances in more detail, it was first investigated whether the same disturbances can also be caused by conducted interference and which sensor line is susceptible to such conducted disturbances. It could be determined that a conducted interference on the clock line of the I²C can cause the same kind of disturbances as those described in [4]. Therefore, in this work, we investigate the malfunctions caused by conducted interference to the clock line. In the following, the used measurement setup, the sensors and the I²C protocol used for communication are described.

2.1 Measurement Setup

The measurement setup shown in Figure 1 was used. First, without an oscilloscope to determine the susceptibility to conducted interference. The oscilloscope was then used to analyze the I²C. A highpass filter was used to keep the I²C signal away from the signal generator. And a lowpass filter was used to protect the controller from the injected noise voltages without affecting the I²C function. Because of the capacitance in the filter, $1\,\mathrm{k}\Omega$ pull-up resistors were used between the I²C lines and $3.3\,\mathrm{V}$. To enable automatic measurements, the signal generator, oscilloscope and controller were connected to a computer. A possibility to reset the controller and the sensor independently from each other has also been provided.

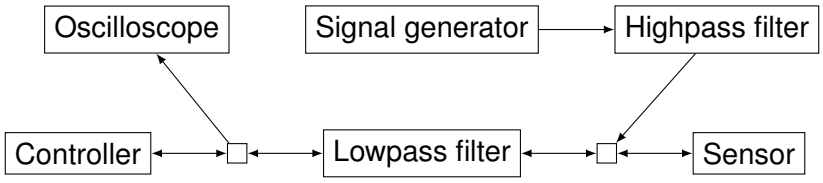

Figure 1: Measurement setup

2.2 Used Sensors

Measurements were made on six different sensors from three different manufacturers (see table 1). All six sensors are a chip which contains an I²C interface in addition to the sensor unit. Five sensors (all except sensor 4) are MEMS which measure the temperature in addition to the actual measured parameter. All sensors were wired according to their corresponding data sheets and provided with appropriate additional capacitors.

Sensor no.	Sensor type	Manufacturer no.
1.	3-axis accelerometer, 3-axis gyroscope	1.
2.	3-axis accelerometer	2.
3.	3-axis accelerometer	3.
4.	temperature sensor	3.
5.	differential pressure sensor	3.
6.	absolute pressure sensor	3.

Table 1: Tested sensors

2.3 I²C

I²C is a bidirectional 2-wire bus developed in 1982 by Philips Semiconductors. It is a widely used bus standard for intra-board communication [1]. The standard data rate is up to $100\,\frac{\text{kbit}}{\text{s}}$. The I²C consists of two lines, the data line (SDA) and the clock line (SCL). Both lines are standard high. The clock signal is always driven by the controller, but a device can stretch the clock by holding down the clock line when needed. A data block consists of $9\,\text{bits}$, $8\,\text{bits}$ of data and one Acknowledge (ACK) bit which is set by the data receiver by pulling low the SDA line [6]. The I²C communication between the controller and the sensors starts with $8\,\text{bits}$ (Sensor I²C adress and a write/read bit) set by the controller, followed by an ACK bit from the sensor. With five sensors two further $9\,\text{bit}$ blocks followed where the data are sent from the controller and specify which data are to be sent from the sensor. This is followed by several blocks in which the sensor sends the measured values and the controller only sets the ACK bit. Figure 2 shows an example of communication between sensor 4 and controller. The transmitter of the bits can be recognized by the different low level. The sensor has a lower low level than the controller.

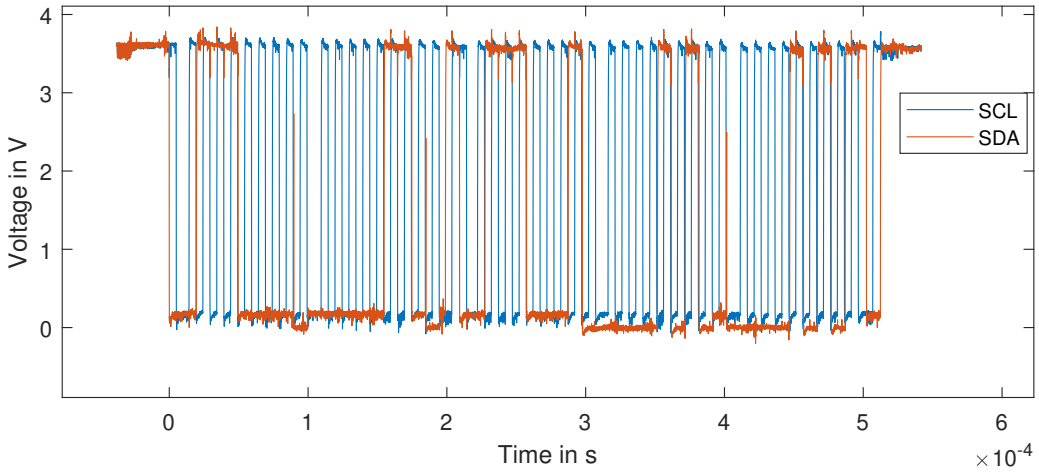

Figure 2: I²C communication sensor 4. Three $9\,\mathrm{bit}$ blocks from the controller and two blocks from the sensor containing the temperature readings.

3 Measurement Results

All six sensors were tested for susceptibility to interference. For this purpose, an interference voltage was applied to the SCL line which was varied in frequency and amplitude. The sensor data received via I²C was monitored and the corresponding measurement data generaterd by an oscilloscope was stored.

3.1 Malfunctions encountered

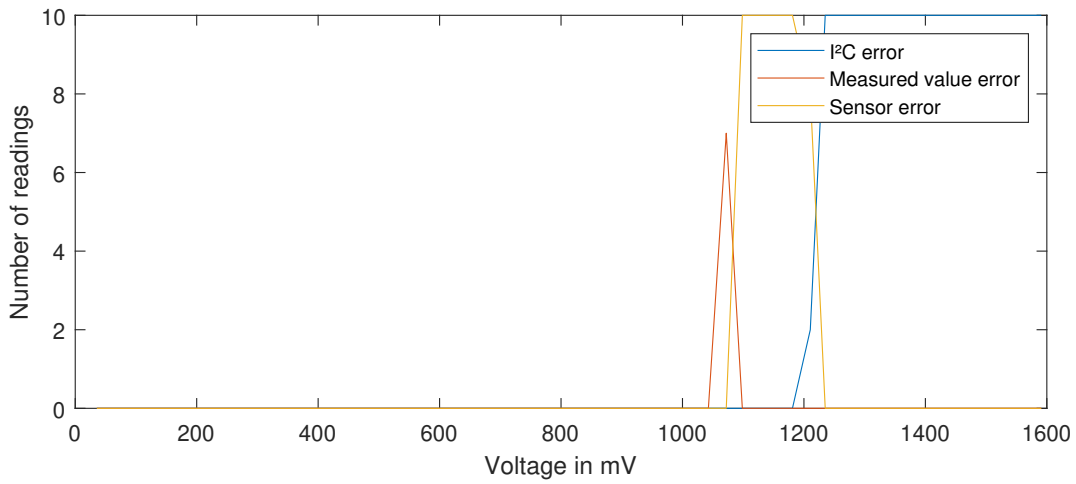

Figure 3: Example of malfunctions at sensor 3 and $500\,\mathrm{MHz}$. With ten readings per voltage step.

The malfunctions that occurred can be divided into three categories. The measured value error where incorrect measured values are received. The controller error where the controller crashes and must be reset. And the I²C error where no complete I²C communication takes place. These three problems were found in all tested sensors. Only one error that occurred with sensor 5 cannot be classified in these categories - a sensor crash that could only be remedied by a sensor reset. No sensor crashes were detected with the other sensors. If the interference voltage is increased at the same frequency, first measured value errors, then controller errors and finally I²C errors occur (see Figure 3). The voltage range and whether they occur at all depends on the frequency and

on the sensor type. Figure 4 shows the occurrence of the I²C error as a function of amplitude and frequency of the interference voltage for 5 sensors. The first occurrence of the controller error is shown in Figure 5. With sensor 2, the errors only occur at very high amplitudes and only at $100\,\mathrm{MHz}$ and are therefore not shown in this figures.

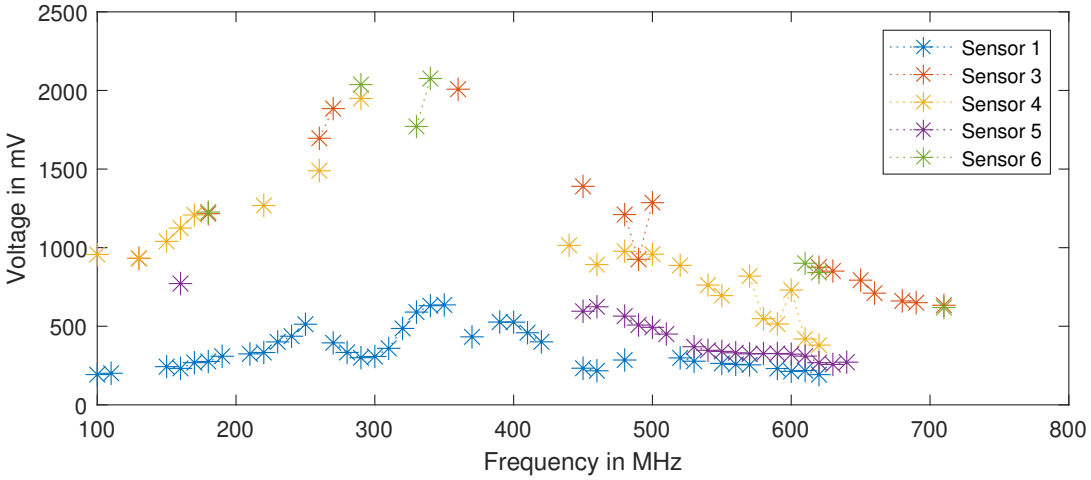

Figure 4: Occurrence of I²C errors

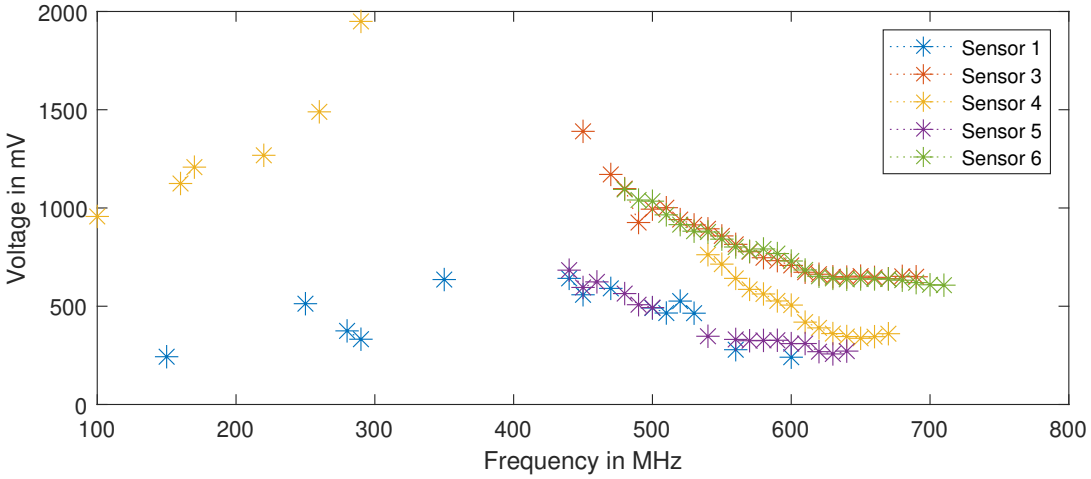

Figure 5: Occurrence of controller errors

3.2 Error cause

The I²C error occurs because the sensor can no longer read the I²C communication. This is simple to detect because the sensor does not set any or all ACK bits.
The reason of the controller errors can be determined from the oscilloscope values, as shown in Figure 6 and 7, for example. In Figure 6 it can be seen that sensor 2 sets its third ACK bit one bit too early. In Figure 7, the second ACK bit was sent one bit too late. Such a too early or too late setting of the ACK bit from the sensor is present with all detected controller errors.

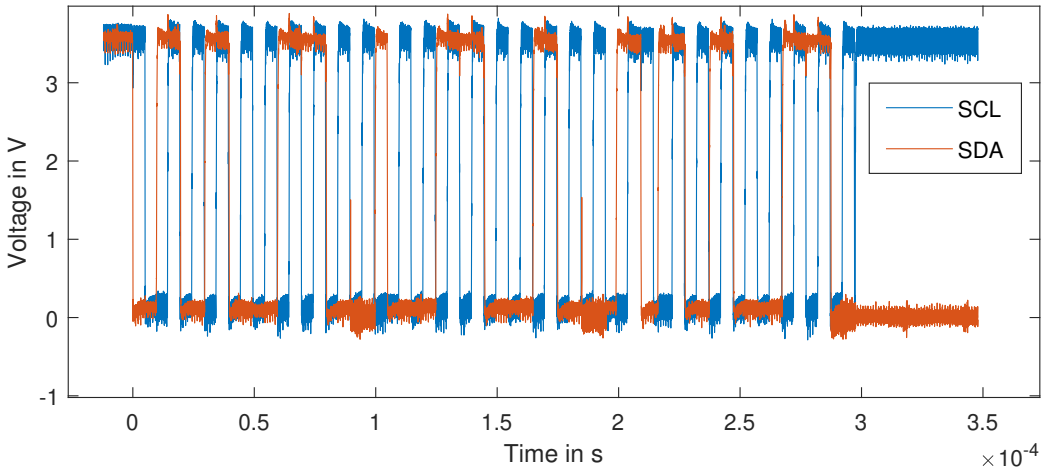

Figure 6: Controller error with sensor 2 and 100 MHz. Third ACK bit at bit 8 of the third 9 bit block. Last bit in the image.

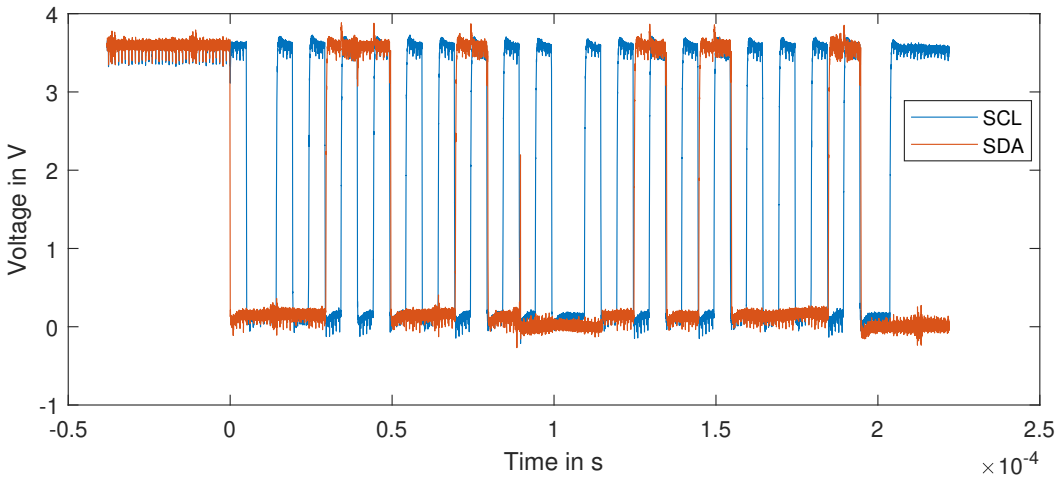

Figure 7: Controller error with sensor 3 and 110 MHz. Second ACK bit one bit too late. Last bit in the image.

In the case of the most measured value errors, as shown in Figure 8, the sensor stops transmitting in the middle of the transmission process. Sensor 1 does not terminate the transmission but sends only 1 bit per 9 bit block, shown in Figure 9. This behavior could be explained by the fact that the I²C protocol allows a slave (the sensor) to switch from write to read mode by not setting the ACK bit by the master. When this happens the slave would only write ACK bits. If a slave is in write mode and would expect an ACK bit already at the 8 bit of a block. The slave may switch to read mode whereby it would send an ACK bit at the 8 bit of each block or it terminates both the write and the read mode.

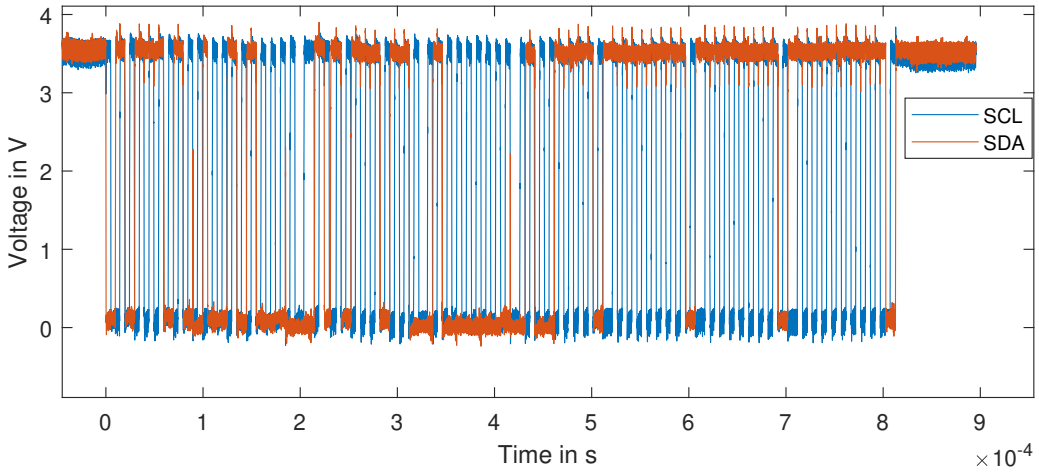

Figure 8: Controller error with sensor 6 and $180\,\mathrm{MHz}$. The sensor stops transmitting after the second data block.

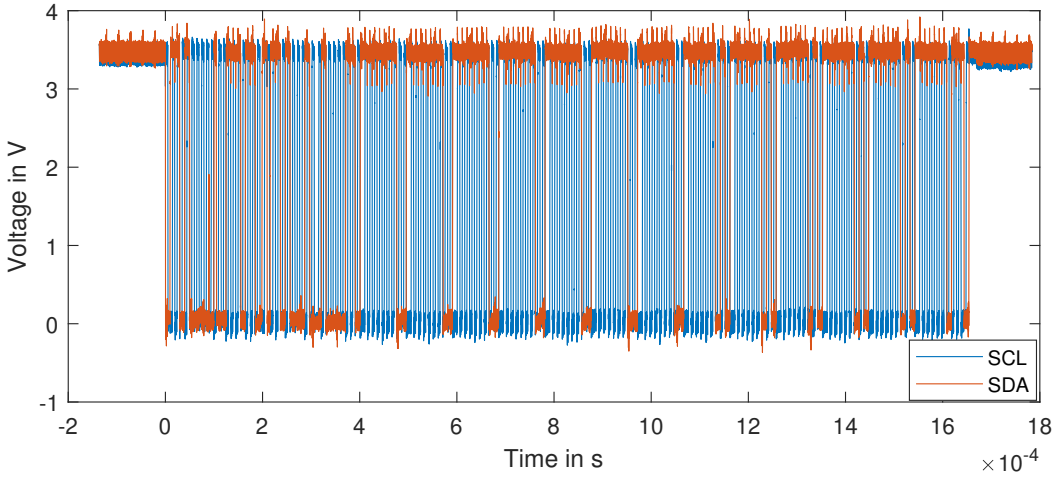

Figure 9: Controller error with sensor 1 and $120\,\mathrm{MHz}$. After the first data block, the sensor sets only the 8 bit of a block and after the 8th data block only the 7 bit.

4 Conclusion and Outlook

Sensor malfunctions, which are caused by the application of interference voltages on the I²C SCL line, were examined and the results were presented. It was shown that all examined sensor systems show the same error types. The sensors are thrown out of sync by applying an interference voltage to the SCL line of the I²C. This leads to the sensor writing or expecting bits on the SDA line at the wrong time. These bits fit perfectly into the bus timing and are difficult to detect. However, they lead to incorrectly transmitted measured values and controller crashes. Depending on the sensor and frequency, an interference voltage at only a few $100\,\mathrm{mV}$ is required to cause these errors.

Both a controller crash and erroneous readings can lead to serious problems in critical systems. Where it would still be possible to fix the controller crashes by hardening the controller against incorrectly set bits, it is only possible to fix the incorrect measured values on the software side if entire measured values are ignored.

By shielding the I²C lines, such interference voltages could be prevented. Filtering the I²C lines in front of the sensors would also be an option, but this could interfere with the I²C function due to the additional capacitances and inductances. An examination of the I²C interfaces of the sensors could also be interesting. Unfortunately, data on the internal structure of such sensor chips is not available.

References

[1] KUMARI, R. Shantha S. ; GAYATHRI, C.: Interfacing of MEMS motion sensor with FPGA using I2C protocol. In: *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, 2017, S. 1–5

[2] LAVAU, Louis C. ; SUHRKE, Michael ; KNOTT, Peter: Susceptibility of Sensors to IEMI Attacks. In: *2021 IEEE International Joint EMC/SI/PI and EMC Europe Symposium*, 2021, S. 533–537

[3] LEACH, P. O. ; ALEXANDER, M. B.: *Electronic systems failures and anomalies attributed to electromagnetic interference*. Bd. 1374. National Aeronautics and Space Administration, Marshall Space Flight Center, 1995

[4] PAHL, Arne ; RATHJEN, Kai-Uwe ; DICKMANN, Stefan: Intended Electromagnetic Interference with Motion Detectors. In: *2021 IEEE International Joint EMC/SI/PI and EMC Europe Symposium*, 2021, S. 324–328

[5] RICHELLI, Anna: EMI susceptibility issue in analog front-end for sensor applications. In: *Journal of Sensors* 2016 (2016)

[6] SEMICONDUCTORS, NXP: UM10204 I2C-bus specification and user manual. In: *User Manual* 7 (2021)

[7] SON, Yunmok ; SHIN, Hocheol ; KIM, Dongkwan ; PARK, Youngseok ; NOH, Juhwan ; CHOI, Kibum ; CHOI, Jungwoo ; KIM, Yongdae: Rocking drones with intentional sound noise on gyroscopic sensors. In: *24th USENIX Security Symposium (USENIX Security 15)*, 2015, S. 881–896

[8] TRIPPEL, Timothy ; WEISSE, Ofir ; XU, Wenyuan ; HONEYMAN, Peter ; FU, Kevin: WALNUT: Waging doubt on the integrity of MEMS accelerometers with acoustic injection attacks. In: *2017 IEEE European symposium on security and privacy (EuroS&P)* IEEE, 2017, S. 3–18