Radar Raw Data Transfer between Radar Sensors and Automotive ECU via SSC

Authors:
Dipl.-Ing. (FH) Marc Steuerer
Prof. Dr.-Ing. Alfred Höß
University of Applied Sciences
Amberg-Weiden

Abstract

This article shows a possibility to connect radar sensors directly to an electronic control unit (ECU) and to transfer sufficiently fast the radar sample data. The approach is part of the German research project AUTOSAFE. The project has been granted by the German Federal Ministry of Education and Research (BMBF). AUTOSAFE aims at the improvement of road traffic safety. The intention is to provide driver support during all driving phases and this has to be realised in a modular, integral safety system. The driver assistance system is clustered in three categories, which are comfort- and safety-oriented driver assistance as well as the combination of active and passive safety systems.

1 Introduction

One sub-project of AUTOSAFE is represented by the development of a collision mitigation system (CMS). This system is a safety-oriented as well as in parts a comfort-oriented driver assistance system. For the perception of the environment, the first development step of the CMS uses three front radar sensors. These radar sensors provide two different bus systems for the communication. One is a CAN bus which is responsible for sending commands to the sensors. The other is the synchronous serial channel (SSC) which is used for the transfer of the radar sample data (see figure 1). Because SSC is much faster and has a simpler protocol than CAN, it is predestined for the data transfer.

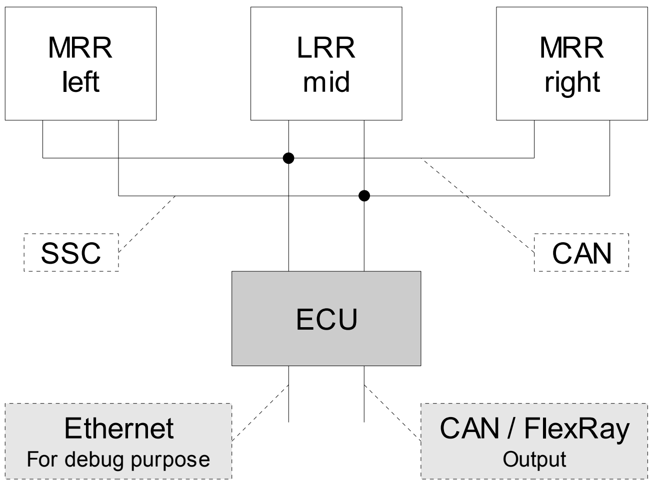

Figure 1: System Overview

Very often radar signal processing algorithms are implemented on a PC and for the data transfer between the sensor and the PC a gateway is responsible. But the advantages to replace a PC by an ECU in automotive applications are on hand: It is more robust, provides a smaller design, the operating system is real time and so forth. Furthermore, this approach reduces the amount of external connections between the hardware components.

The drawback of this approach is that the radar signal processing runs within the ECU and the user cannot debug and display the results of this process as easy as on a PC. For that reason, the ECU provides an Ethernet connection as well. Therewith it is possible to transfer all relevant information and intermediate steps of the signal processing to a PC where it is easier to display this information.

2 Data transfer

For the data transfer it is very important that the used bus system is able to transfer all incidental data in almost real time. The amount of data during a radar measurement is very high. On this account a bus system with a very high performance is needed. The SSC – bus introduced in this paper is such a system.

SSC is a fast synchronous master-slave bus system. Physically, it is a four wire twisted pair connection, were two wires are for the clock and two for the data. The signals are complementary, the voltage levels are between $\pm 2\,Volt$ and the bit rate is $10\,MBit/s$.

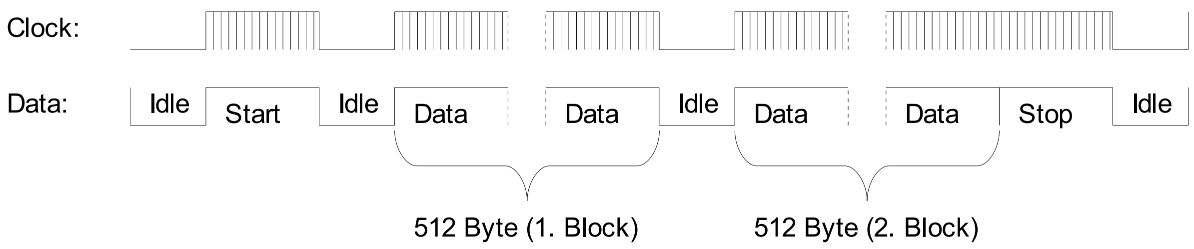

Figure 2: SSC - Protocol

The used protocol is quite simple (see figure 2). It is composed of a start- and a stop sequence with two bytes each, the data blocks and a couple of idle times. During the idle times the clock is also logical zero. The first data block contains the samples of the up sweep and the second data block the samples of the down sweep. Each measurement for the up- and the down sweep respectively has a length of 256 samples. The resolution of each sample is 16 bits, thus two bytes. Additionally to the payload data the protocol needs four bytes for the start- and stop sequence. Altogether the sensors have $512\ bytes \cdot 2 + 4\,bytes = 1028\ bytes$ to transfer.

The transmission of one byte takes $10^{-6} \cdot 8 = 800\,ns$. The approximately amount of the idle times is $20\,\mu s$. With this information the total transmission time can be calculated.

$$1028\,bytes \cdot 0.8\,\mu s + 20\,\mu s = 842.4\,\mu s$$

The transmission time of the pure payload data (without the idle times) takes $822.4\,\mu s$, so the ratio between the pure and the total transmitted information is:

$$\frac{(822.4\,\mu s)}{(842.4\,\mu s)} = 0.98$$

This again is equivalent to 98 percent. In comparison with other bus systems, in particular the CAN bus where the ratio is about 60 percent, the ratio of the SSC – bus is very good.

For the triggering of the radar sensor it is necessary to send some control information via the CAN bus. Three CAN messages have to be sent for the whole measurement and data transfer process. This procedure is illustrated in figure 3. The used CAN bus is a high speed bus and works with a speed of $1\,MBit/s$. The length of the messages is four bytes in all cases. That leads to an approximate transfer time of $78\,\mu s$ per message. This CAN network is composed of just the radar sensor and no other clients are connected to it.

Meaning and progress of the CAN messages are as follows. First, it is necessary to send a command which starts the measurement, next step is to set the sensor as master. This means the sensor is responsible for the generation of the clock and for the preparation of the radar data. If the master set command was successful, the sensor answers with a positive reply. Now it is possible to start the data transfer via SSC, this happens because of the third CAN message.

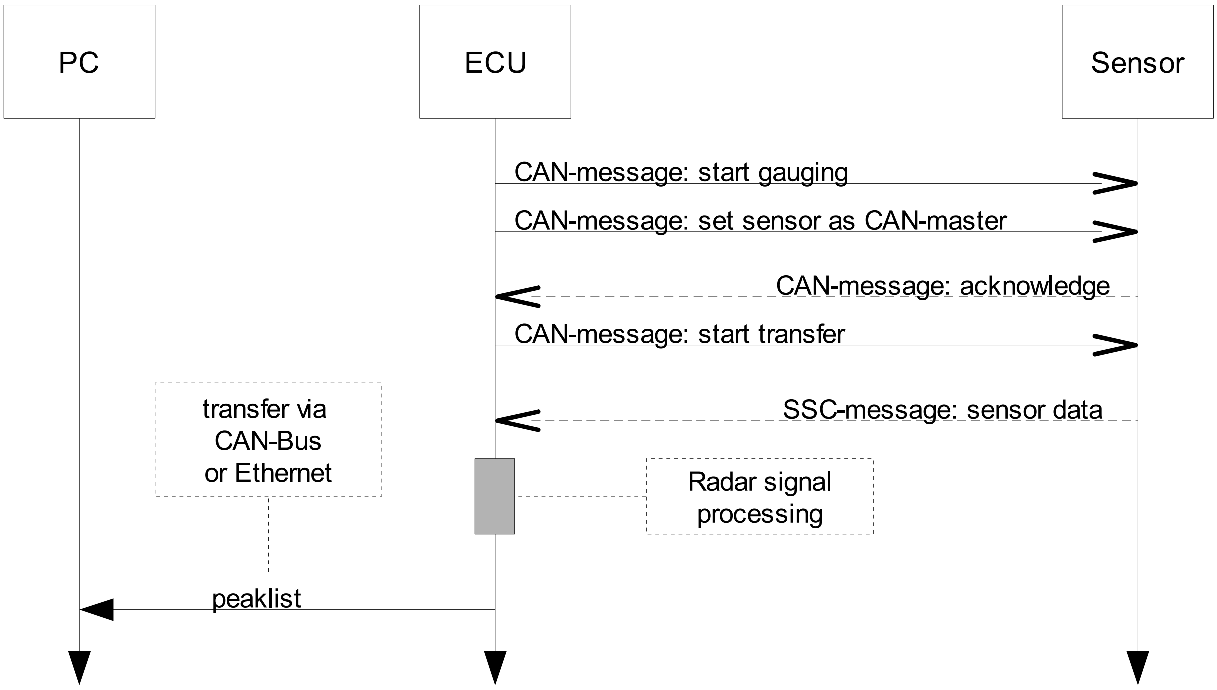

Figure 3: Data transfer

After the transmission of the raw radar data via SSC, all necessary information was transferred to the ECU and the signal processing can start. The results as well as intermediate steps can be transmitted via the CAN bus, Ethernet or FlexRay to other ECUs or for debugging and displaying to a PC. In figure 3 the transmission of the generated peak list to a PC is exemplarily illustrated.

3 Realisation

For buffering the radar raw data within the ECU, a hardware FIFO buffer is used. The size of this buffer is 512 bytes. Since the sensors transfer 1028 bytes it is necessary to clear the buffer during the transmission process and to store the information into an internal memory (software double buffer). For reading the data out of the hardware FIFO automatically a so called “alarm level” is responsible. If the alarm level has sent a trigger impulse to the software, it starts to acquire the data. During the read out process it is possible to fill the FIFO further. It is very important to set the alarm level to a suitable value. For example, if the alarm level value is too high, the FIFO buffer could overflow before the software has readout all entries.

For the calculation of the alarm level several items need to be considered. The FIFO buffer has to be read out automatically with the last incoming data byte from the sensor and the FIFO must not overflow. The value of the alarm level has to be in the range of the FIFO buffer size.

An integral divisor is needed which allows to read out the FIFO after the very last transmitted byte. The approach in this matter is the prime factor analysis. That leads to the following result:

$$257 \cdot 2^2\, bytes = 1028\, bytes$$

From this it follows that the desired value for the alarm level is 257. That means after every 257 bytes the software reads out the FIFO buffer automatically.

From figure 2, one can conclude that this value represents a good choice. The up sweep contains

514 bytes: the down sweep contains the same number of bytes. With the chosen alarm level, the software has to clear the FIFO buffer twice for the up- as well as twice for the down sweep. The idle time between the up- and the down sweep gives the ECU a little bit more time to readout and store the information.

But what happens if some data get lost during the transmission process? This case has to be considered and checked. If this case occurs, the received data are not valid and have to be discarded. To solve this problem a timeout is used. If this timeout has expired and the FIFO buffer has not been read out four times an error has occurred during the transmission process. It is mandatory to choose the time for the timeout marginally larger than the total transmission time.

If the transmission was not valid and the data have been discarded, a new transmission process has to be started automatically. It does not make any difference, whether the error occurs in the up sweep or in the down sweep. A completely new measurement and transmission process has to be started.

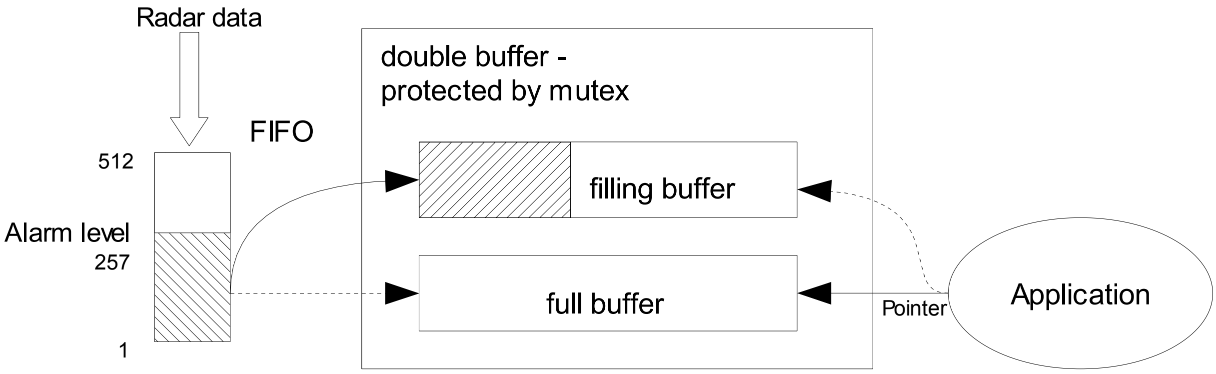

Figure 4: FIFO system

The valid data are stored in a double buffer (see figure 4). With this kind of buffering the data have to be provided to the application. One buffer stores the next incoming data packet, whereas the other buffer provides the data to the application. But before the data is transferred to the double buffer it is necessary to remove all irrelevant bytes from the incoming data stream. These are the start- and stop sequence.

The signal processing part, which uses the data from the radar sensors, gets just a pointer to the accordant buffer. This pointer has been set automatically to the desired and correctly filled buffer. To avoid simultaneous use of the filling buffer, it is mandatory to use mutual exclusion (mutex) to protect this access. Therewith it is impossible that the application gets access to a buffer which is in the filling process at the moment.

4 Conclusion

The SSC – bus is a suitable interface to transfer raw radar data from the sensor to an ECU. Therewith it is possible to send the requested data almost in real time and the latency time between the first request and the start of the data transmission is also satisfying.

A big advantage is that the data transfer and the whole signal processing are running on just one ECU and not on different systems. Further on, it avoids problems which can occur with different transmission lines and with the use of PCs in automotive applications (see also section 1). Anyway, it is possible to send important information to a PC or show signal processing steps on a display which is plugged to the ECU.

5 Acknowledgement

This work was supported by the German Research Ministry (BMBF). The authors want to thank the Nanoelectronic unit of the BMBF for granting AUTOSAFE under funding no. 01M3076.

6 References

[1] Tobias Beyrle, Diploma Thesis, Entwurf und Realisierung einer Anbindung von Radarsensoren an eine ECU mittels SSC- und CAN Bus, to be published in 2007

[2] Robert Bosch GmbH, Techical Report, CAN Specification, 1991

[3] Freescale Semiconductor, Manual, MPC5200B User's Manual, 2006

[4] Karl-Heinz Mattheis and Steffen Storandt, Manual, Arbeiten mit C166-Controllern, 1995