CHALMERS
UNIVERSITY OF TECHNOLOGY

Diagnostics on Gigabit Speed SerDes Video Links

Master's thesis in Master Programme Systems, Control and Mechatronics

SIMON LEHTONEN

Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2017

Master's thesis 2017:XX

Diagnostics on Gigabit Speed SerDes Video Links

SIMON LEHTONEN

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Signals and Systems
Division of Communication Systems
Chalmers University of Technology
Gothenburg, Sweden 2017

Diagnostics on Gigabit Speed SerDes Video Links
SIMON LEHTONEN

Supervisor: Henrik Fagrell, CEO, Ph.D, Diadrom Systems AB
Advisor: Anver Hisham Unnichiriyath Siddique, Ph.D Student, Department of Signals and Systems
Examiner: Thomas Eriksson, Professor, Department of Signals and Systems

Master's Thesis 2017:XX
Department of Signals and Systems
Division of Communication Systems
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2017

Diagnostics on Gigabit Speed SerDes Video Links
SIMON LEHTONEN
Department of Signals and Systems
Chalmers University of Technology

Abstract

The fast paced development of Advanced Driver Assistance Systems (ADAS) in the automotive industry, boost the demand for ways of ensuring safe and reliable systems. With the increased data speed of video links, connecting the high resolution cameras with their Electrical Control Units (ECU), ADAS are becoming more sensitive to electromagnetic disturbances. Due to this sensitivity, the conditions of video links in the car should be investigated continuously to ensure the safety and reliability.

The outcome of this thesis is the concept Simons Video Interference Check (SIVIC), which is designed to be a technique for diagnostics on video links. The SIVIC concept is able to continuously record video link quality data, detect trends of deteriorating behaviour, detect and distinguish multiple types of disturbances as well as detect video link failures. The SIVIC concept was implemented as a proof of concept tested on two different test rigs based on the technology Gigabit Multimedia Serial Link (GMSL). One of the test rigs contained a parking assist camera, which is developed to be included in a vehicle and representing a real system used in the automotive industry. The video link failure detection was tested by real world tests in a lab at Research Institute of Sweden (RISE) and all other functionality were tested by using software. The SIVIC concept proves how a software module can be used to detect and predict data link failure of high speed automotive video links.

Keywords: ADAS, ASIL, SIVIC, CML, EMC, GMSL, SerDes, LVDS, UART

Acknowledgements

I would like to express my gratitude to Diadrom for providing me with the opportunity to perform my master's thesis at their company in a welcoming and inspiring environment. A special thanks to my supervisor Henrik Fagrell, Jonas Hellberg, Andreas Larsen and Tobias Rasmusson for your valuable input and support during the process. I would also like to thank Mikael Seger at RISE for letting me use their lab and equipment and Matthew Coleman at Maxim Integrated for supplying me with hardware and helping me with the test rigs.

My academic supervisor Anver Hisham has been very helpful in giving me feedback on my report and I would like to thank him for that.

Simon Lehtonen, Gothenburg, 06 207

Abbreviations

ADAS	Advanced Driver-Assistance Systems
ASIL	Automotive Safety Integrity Level
CML	Current Mode Logic
CSV	Comma Separated Values
ECU	Electronic Control Unit
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
GMSL	Gigabit Multimedia Serial Link
HDMI	High-Definition Multimedia Interface
LVDS	Low Voltage Differential Signalling
OSI	Open Systems Interconnection
PRBS	Pseudo Random Bit Sequence
SerDes	Serializer/ Deserializer
SIVIC	Simon's Video Interference Check
STP	Shielded Twister Pair
UART	Universal Asynchronous Reciever/Transmitter
USB	Universal Serial Bus

Contents

List of Figures

List of Tables

1
Introduction

All major car manufacturers are focusing on developing autonomous driving and developing active safety features to include in their future car models. This means the need for data links for cameras and displays is rapidly increasing. A handful of high end cars in today's market are already equipped with more simple versions of autonomous driving components and these features are rapidly becoming more common for manufacturers to offer as an addition while selling a car.

These systems enables great capabilities but also bring forth some issues. A common question asked by people in the industry is about the responsibility in case of an accident, and there is no definite answer for that. For now, the driver of the car is liable for the possible accidents. But in the future, with changes in the regulations, the responsibility might shift to the car manufacturers. We have to wait and see.

Safety is a main focus in these systems and there are standards such as ISO26262 [10] which contains requirements for how the electrical, electronic and software should be developed for building safe systems. These standards are highly important while developing autonomous driving systems, especially since the complexity of them is ever increasing. The manufacturers must be able to guarantee that every part of the system is working properly and must ensure the safety. This thesis is about the system safety, and covering all parts of the system. This also includes the high speed data links between the subsystems and camera applications. There is also a belief that the high speed video link is overlooked. The camera is one of the main sensors used for autonomous drive and it is crucial to know its performance. Therefore diagnosis of the data link performance is necessary. The camera systems are however covered in the ASIL requirements, hence the manufacturers of high speed data links have included functionalities enabling further diagnostic applications [11].

With the increased usage of high resolution cameras, the need for data links capable of handling large amount of streamed digital video data has grown. The manufacturers strive to minimize the cost and are required to fullfill standards such as ISO26262 [10], steering them to use automotive specific video interface solutions instead of common consumer video interfaces such as High-Definition Multimedia Interface (HDMI). There is no shortage of video interface solutions and all of them offer something which is worth considering while selecting one. It is common for these interfaces to be based on current mode logic (CML) in the physical layer, which transmit the video data over a serial link by using a serializer and deserializer. This is called SerDes interfaces [11].

Inova Semiconductors APIX, Maxim Integrateds GMSL and Texas Instruments FPD-link are all similar versions of SerDes interfaces designed to be implemented in an automotive environment. All of these SerDes interfaces contain three main components, a video channel, a control channel and power transmission to the camera or display. All of these components can be transferred over a single four wire or two wire shielded twisted pair (STP) cable which brings down the costs and avoid any need for multiple cables. In other solutions, three separate cables may be needed for the same purpose. Both Inova Semiconductor and Maxim Integrated have announced that as of the year of 2017, they are going to release new versions of these protocols. Both of these are specific to handle link speeds up to 12 Gbit/s enabling even more advanced sensors in the future at the cost of being more sensitive to the disturbances.

This thesis uses hardware based on the Gigabit Multimedia Serial Link (GMSL) protocol. The main reason for this, is that Diadrom System AB already had an established connection with Maxim Integrated and also some evaluation boards from them. Due to the time constraints, the scope of this thesis is limited to only one technology.

1.1 Purpose

The purpose of this thesis is to explore if it is possible to build a diagnosing software for analysis of high speed data links in future vehicles. The need for this kind of software will also be investigated.

1.2 Research question

The main research question is:

- How can a software module using SerDes signals detect and predict data link failure.

By narrowing the problem the following topics are considered;

- What kind of disturbances occur in a car that will affect the high speed data links?
- Is it possible to inflict disturbances on a CML link in a lab environment that would emulate the real life disturbance behaviour?

1.3 Delimitations

- The thesis only consider GMSL based data links from the semiconductor manufacturer Maxim Integrated [12].
- The thesis will only consider the built in fault detection parameters in the GMSL serializers and deserializers while collecting and analyzing data.
- The thesis only focuses on creating a software to use on the test hardware. The software is not built to be implemented in a car, instead it is to be seen as a proof of concept.

2

Research Methodology

2.1 Design Science Research

This thesis project is realized based on the design science research [1] which is an iterative methodology for applying research on information systems. This method contains five steps which are revisited throughout the project, which are depicted in figure 2.1.

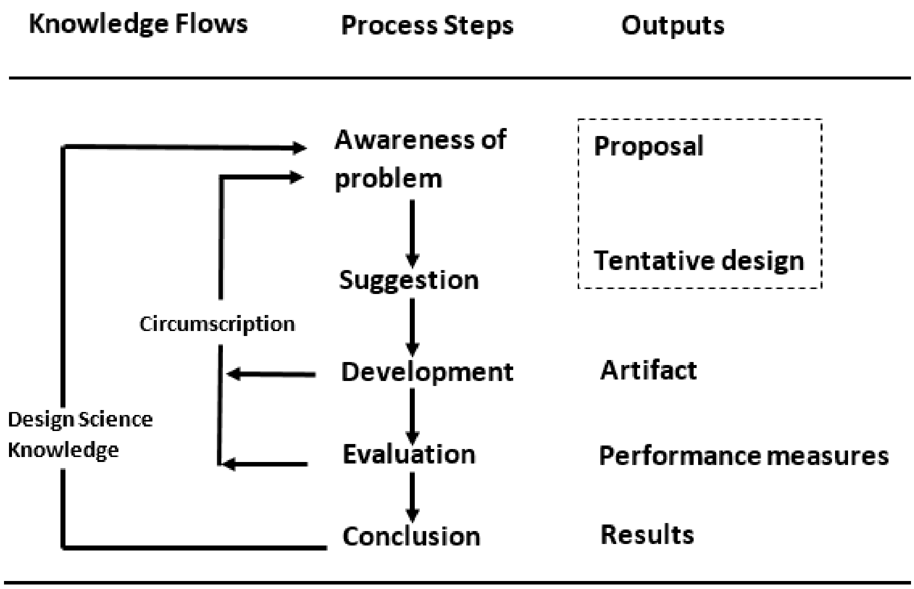

Figure 2.1: Design science research methodology [1].

The first step in this method is called *Awareness of the problem*. Here the researcher becomes knowledgeable about an interesting topic which is then formulated into a proposal for a new research endeavor.

The second step in this method is called *Suggestion*. Here the researcher uses their current knowledge to form an experimental design with the purpose of solving the problem from the previous iteration. This design can be modified in future iterations depending on how the project evolves.

The third step in this method is called *Development*. Here the researcher implements the experimental design and creates an artifact based on it. The researcher often gains new knowledge during the implementation and it is possible to return to the previous steps for further improvement of the proposal and experimental plan based on the recently gained insights.

The fourth step in this method is called *Evaluation*. Here the researcher evaluates if the artifact developed in the previous step solves the research problem from step one. Possible deviations from the proposal have to be noted and explained. If the researcher is unsatisfied with the result it is possible to go back to the previous steps to reform and improve the proposal and experimental plan. These adjustments are based on the new knowledge gained by the researcher during the course of the project.

In the final step *Conclusion*, the researcher studies the results of the research project and evaluates if it can be generalized and applied to similar situations. The researcher also analyses the results to see if it can be the basis of a proposal for a future research endeavour.

2.2 Literature Studies and Experiments

To follow the methodology according to the design science research [1], the researcher has to have knowledge about the topic at hand. To develop an understanding for the area of low voltage differential signalling (LVDS) based communication and advanced diagnostics, it is necessary to carry out literature studies in these subject areas. The initial part of this thesis project contains literature studies on topics such as advanced diagnostics, LVDS, Universal Asynchronous Receiver/Transmitter (UART), Electromagnetic Compatibility (EMC) and disturbances in an automotive environment. This has built a foundation of knowledge which was later used to develop an experimental design in the suggestion step as well as implementation in the development step of the design science research method.

Literature studies are not enough to completely understand something. It is often helpful to perform some experiments on a system to study its behaviour. Parallel to the literature studies in the initial steps of the thesis project, some experiments on the test hardware were also issued. These experiments were helpful in creating a viable set of requirements for the suggestion phase of the design science research in section 2.1.

Experiments were also used in the evaluation phase for evaluating the quality of the artifact generated in the development phase as well as for collecting a data set used during the development of the artifact.

3

Technical Background

3.1 GMSL

In the year 2008 Maxim Integrated [13] released Gigabit Multimedia Serial Link (GMSL) as a communication link for video applications in the automotive industry. GMSL is based on SerDes technology which means that it uses a serializer on the transmitter side and a deserializer on the receiving side. It is specifically designed to be used for Advanced Driver Assistance Systems (ADAS) and Camera Monitoring Systems (CMS) purposes [12]. It is able to provide video transfer speeds up to 3 Gbit/s and it uses a STP or coaxial cables which are both inexpensive and very robust for EMC disturbances. On the physical layer of the OSI model GMSL uses CML.

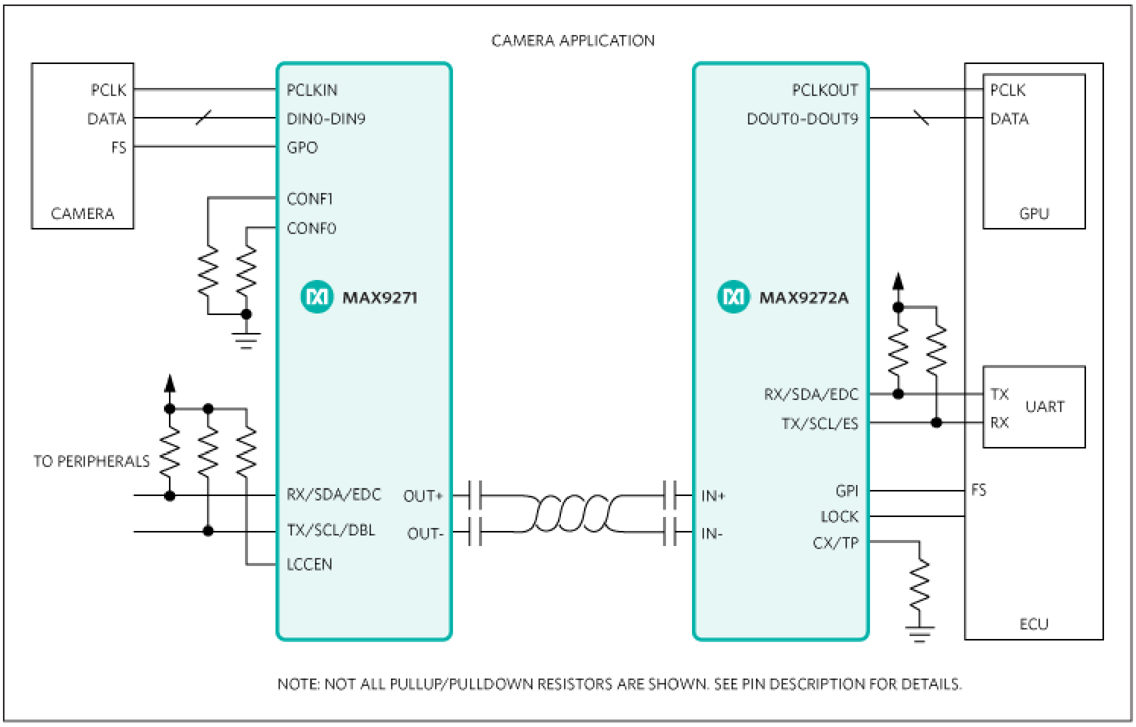

Figure 3.1: GMSL block diagram from a camera application. MAX9271 is the Serializer device and MAX9282A is the deserializer device in this picture. The picture is from [2].

GMSL includes a simplex video channel based on SerDes 8b10b and a duplex control channel based on UART or I^2C. The control channel is used for reading and writing registers in both the serializer and deserializer devices. Even though the video data is simplex, the control channel can send data both ways. GMSL is able to include both the video channel and the control channel on a single STP cable, moreover it is also able to supply a device with power over the same link. This is a key selling point for this system since the car manufacturers always strive to cut down the cost and weight. The serializer scrambles the video data with the forward control data and transmit them in the same frame. On the receiving end, the deserializer de-scrambles and decodes the data, then separates it into video data and the control channel data again [8].

3.2 CML

Current Mode Logic (CML) is a widely used standard for serial signaling on physical layer devices for data speeds more than 1 Gbit/s [3]. CML uses differential signaling and the transmitter includes a differential pair with a 50 Ω resistor on each line. It has a voltage swing of 800 mV and is able to reach high data rates because of the output transistors being constant in the active region which enables fast switching.

The 50 Ω resistors also add termination of the source while driving transmission lines with 50 Ω resistance. This limits the possible reflections in the transmission [3]. Figure 3.2 presents a simple diagram of a typical CML communication system.

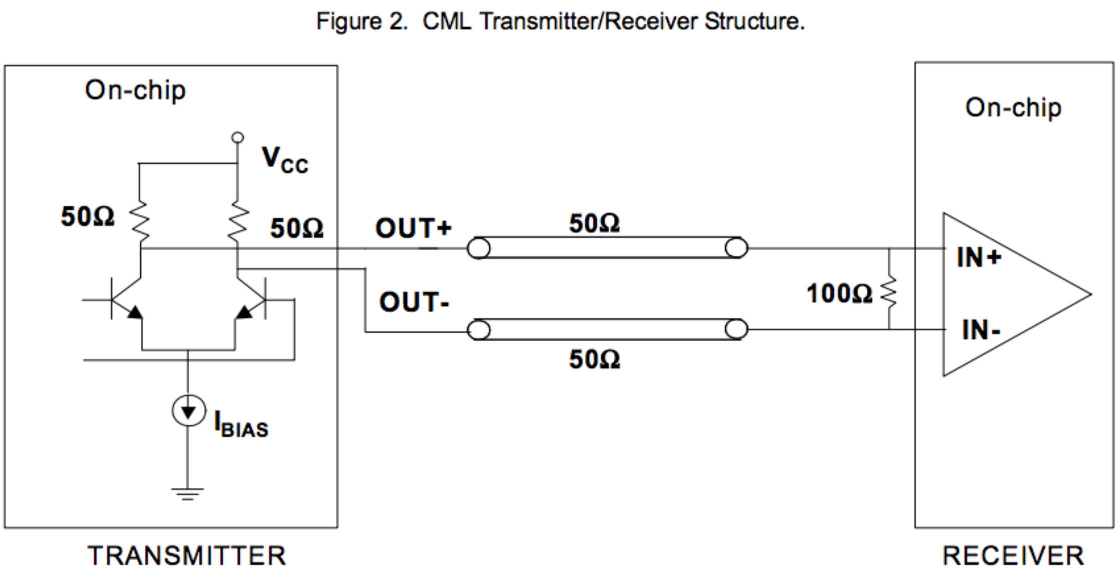

Figure 3.2: Typical structure of a CML transmitter and reciever from [3].

3.3 SerDes communication

Serializer/Deserializer (SerDes) are devices used to transmit a parallel bit stream over one or a few high speed differential serial links by converting it into a serial bit stream [4]. The transmitter side of the communication houses the serializer and is responsible for compressing the parallel bit streams before sending over the differential serial link(s). The receiving end houses a deserializer and restores the serial bit stream into parallel bit streams again.

There exists multiple varieties of SerDes interfaces that use different clock implementations [14]. However, only SerDes using 8b/10b encoding will be covered in this thesis. The encoding 8b/10b works by adding two bits to each data byte creating a 10 bits symbol. The extra bits are used to make sure that the difference between the number of ones and zeros stays within the requested range for a charge balanced data stream [15]. Figure 3.3 illustrates the basics of a SerDes communication using 8b/10b encoding where two extra bits are added to the data byte before the transmission and then removed before deserializing the data byte.

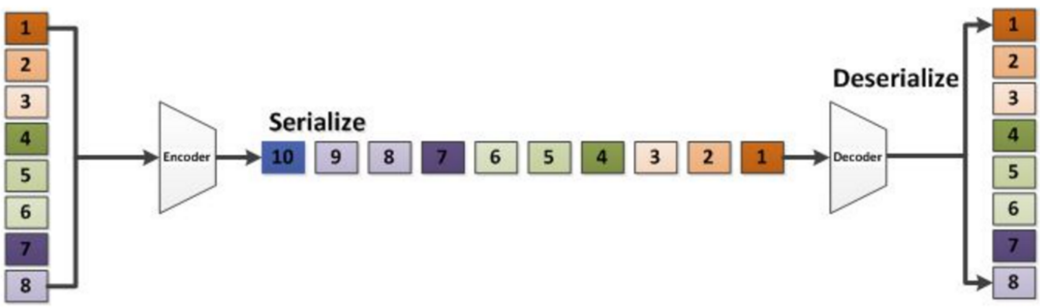

Figure 3.3: Overview of SerDes using 8n/10b encoding from [4].

SerDes interfaces are widely used in automotive video applications and semiconductor manufacturers such as Maxim Integrated [13], Inova Semiconductors [16] and Texas Instruments [17] all have thier own versions which are called GMSL, APIX and FPD-Link respectively. They are all based on CML technology and consist of a video channel and an embedded control channel [11].

3.4 UART

Universal Asynchronous Receiver/Transmitter (UART) is a protocol for serial communication used in computers and other electronics. UART uses SerDes and since it is asynchronous, i.e., it is not dependent on a clock signal from the receiver, instead the timings of the communication are decided right before the transmission of the data [18]. UART uses start and stop bits to each data frame which marks the beginning and end of each word. The start bit notifies the receiver that a data frame is to be sent and synchronizes the clocks of the receiver and transceiver which enables the communication. At the end of the transmission, the stop bit notifies the receiver that the transmission has been completed. If the receiver does not obtain a stop bit, it labels the transmitted data as faulty and discards it. A loss of a stop bit could be due to asynchronous clocks or external disturbances [18].

Since UART synchronizes the clocks of the devices with each data frame, the transmission line can be idle when there is no data.

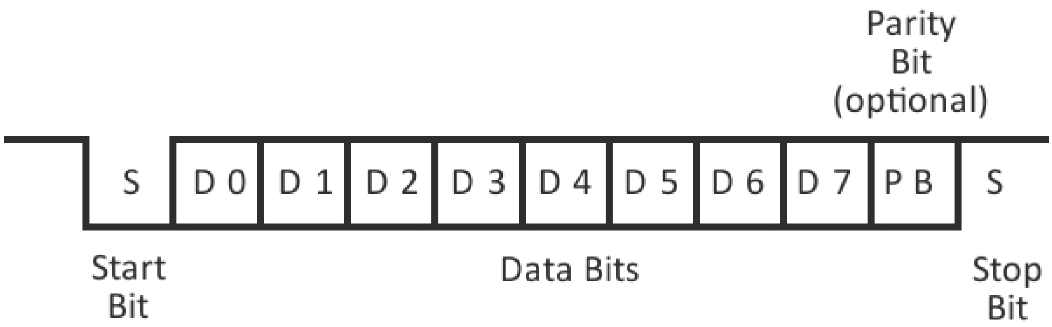

Figure 3.4: UART data frame with start and stop bit from [5].

3.5 Equalization and Preemphasis

Equalization and Preemphasis are tools for compensating for signal deterioration [19]. Preemphasis is processing on the transmitting side and equalization is processing on the receiving side. Both are basically ways of filtering the signal to compensate for the signal deterioration caused by the cable, thereby lowering the bit error rate [20].

3.6 Eye Pattern/Eye Diagram

Eye pattern measurements is a tool for measuring how good a link is at transferring data at the physical layer of the OSI model [21]. An eye diagram is formed by taking the sum of the samples from the high speed digital signal and superimposing the one and zero bits and the associated transition measurements such as rise and fall times. Eye patterns can be used to measure jitter on the link as well as data reflections caused by inadequate termination. It is the opening of the eye which determines the link quality. For optimal performance it would be 100% open, and when the link is exposed to disturbances, the opening will decrease making it harder for the system to distinguish the rise and fall edges of the signal.

Figure 3.5 is an example of how an eye pattern diagram might look like. The horizontal eye opening is used for measuring the amount of jitter in the signal. The wider the horizontal eye opening is, the less jitter is in the signal. Jitter can also be measured by looking at the thickness of the line of signals. The vertical eye opening is used for measuring the difference between one and zero bits. The wider the vertical eye opening is, the easier it is to detect the difference between ones and zeros [6].

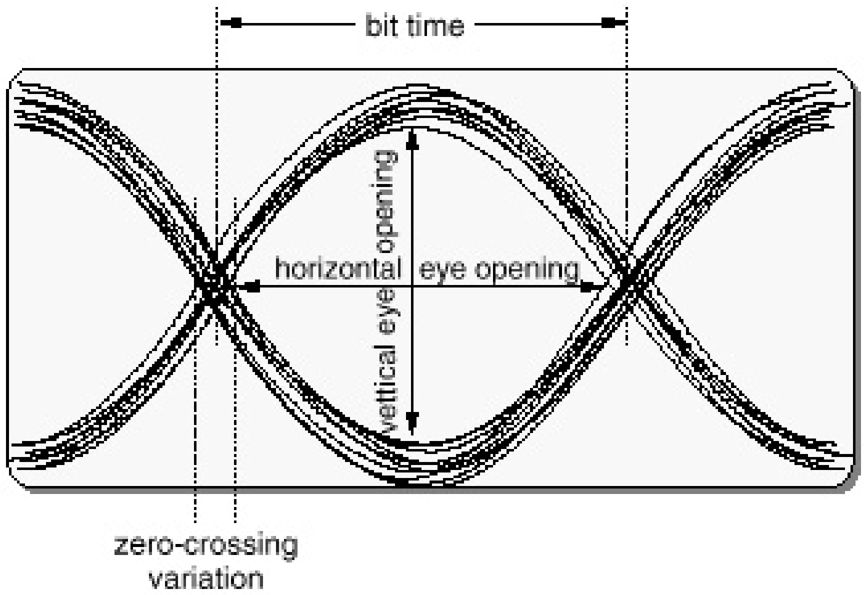

Figure 3.5: Eye pattern diagram from [6].

3.7 Diagnostics

Diagnostics from a software developer's perspective covers development, maintenance and upgrading of software components. A diagnostic software for an after market implementation could include functions for reading and writing error codes, built in tests for verification, calibration and software download [16]. Diagnostics cover many areas and in the automotive industry there is a large demand for diagnostic solutions in order to verify that all safety requirements and regulations are covered. The application ADAS require some diagnostic components in order to fulfill the requirements set by standards such as ISO26262 [10].

3.8 OSI Model

The Open System Interconnection-model (OSI) was developed to define a unified way for communication between computers and other electrical devices. It covers how devices should communicate with each other, ways of ensuring correct data rate as well as ensuring that the data is received by the intended recipient [7]. The OSI model consists of 7 layers wherein the user interactions occurs in the top layer, known as the application layer. The bottom layer, the physical layer, is where the electrical signalling occurs. The OSI model covers the entire range of computer communications and Figure 3.6 presents the different layers of the OSI model with a brief description of each layer.

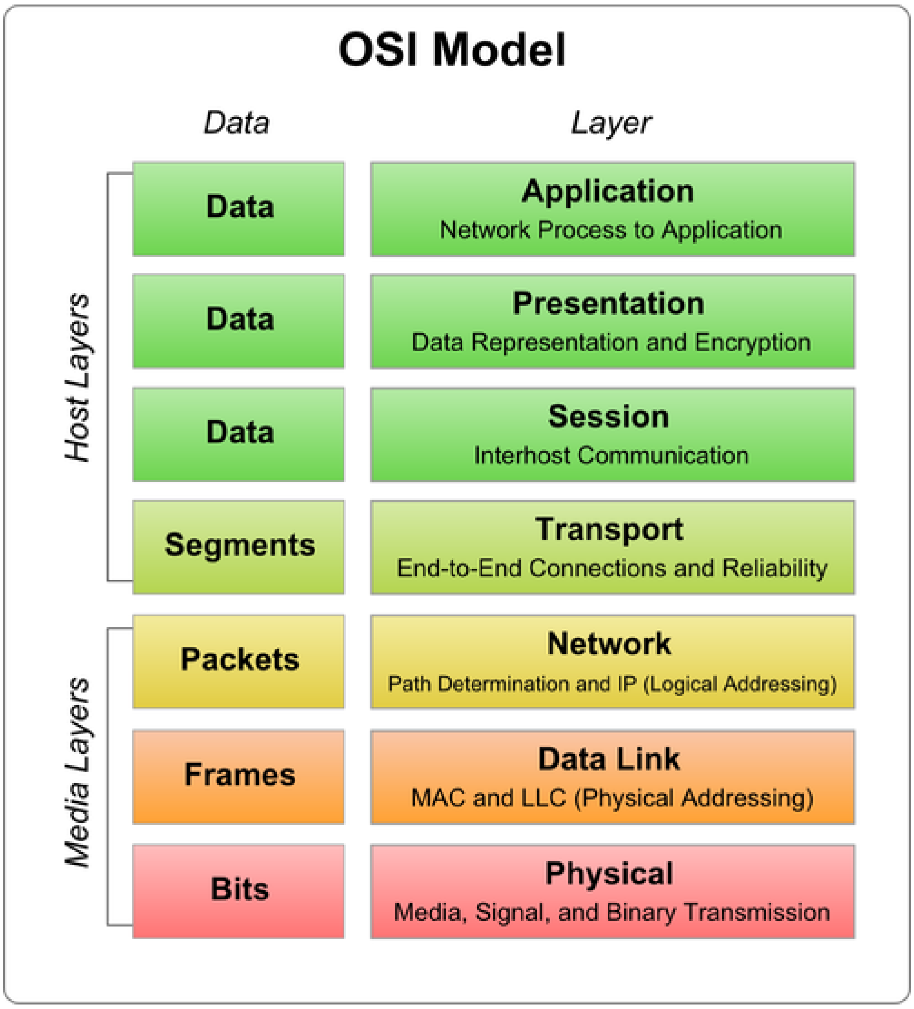

Figure 3.6: The OSI model from [7].

3.9 EMC and EMI

The concept of Electromagnetic compatibility (EMC) and Electromagnetic interference (EMI) should be first reviewed before moving on to the specifics of disturbances that affect CML interfaces.

All implementations of electrical systems and equipment are influenced by the two elements of EMC; emission and susceptibility. Both electrical systems and equipment radiate electromagnetic energy themselves and also pick up electromagnetic energy from other sources. During this process, errors may occur in the circuitry that may lead to system failure. EMC is thus an important part of the electrical system design for limiting the emissions and susceptibility.

There are mainly two types of emitted interference: continuous interference and transient interference. Continuous interference covers signal sources which may emit EMI for long periods of time, but often at a low magnitude. Some examples of these disturbance sources could be television, radio receivers, medical equipment, computers and cellphones. On the other hand, transient disturbances are short, large magnitude disturbances which could be generated by, lightning, electrostatic discharges or electrical motors. Electrical motors often causes repetitive short burst pulses.

Since electronic equipment are ubiquitous in the modern world, it is essential that all electrical systems are designed to radiate as little electromagnetic energy as possible to reduce the impact of these systems on one another. There are regulatory bodies who develop standards and guidelines regarding for EMC requirements. Some of them are IEC, CISPR, FCC, and ISO. ISO regulations and requirements for the automotive industry are the most relevant for this thesis.

Cars have a total emission limit which must be fulfilled by all systems together. It requires extensive testing as well as EMC limited designs for each subsystem.

3.10 Disturbances in an automotive environment

A modern car contains a vast network of microprocessor circuits and sensors with communication links in between. These electrical systems create radiated and conducted emissions which they are also susceptible to. EMC is thus an important topic while designing these systems in order to make them work and to follow the regulations stated by standards such as CISPR25. Hybrid and electrical vehicles contain even more disturbance sources than a conventional car.

The electrical systems are designed to handle these transient and continuous disturbances and should be able to work without being affected by the surrounding systems. However, if there are malfunctions on some systems, for example, a broken shield of a data link, the emissions could corrupt the transmitted data and cause errors in that way.

4
The SIVIC Concept

In this chapter, Simon's Video Interference Check (SIVIC) concept will be presented. This concept is the result of an extensive literature study covering diagnostics of high speed video link implementations in an automotive environment. The SIVIC concept consists of a set of requirements, which when they are met, provides a solution to the research question stated in section 1.2. To meet these requirements, a general diagnostic software (from now on called the SIVIC application) for GMSL video links is developed. This software could in future be the basis for building a car specific implementation for similar purposes. The requirements are a result of the suggestion and development step in the design science research explained in section 2.1.

The electronic control units (ECU) used in cars often have a very limited performance leaving no room to make advanced calculations during normal operations. As a diagnostic component for high speed video links could be quite memory and processing heavy, a solution could be to extract the more advanced tasks to be executed at some other place or time. For the SIVIC concept, the more advanced tasks will be performed after shutdown of the video link, simulating the shutdown of a car. Since the normal tasks and operations of the ECU is not running after shutdown, the SIVIC application is able to use more hardware processors. This separates the SIVIC concept into two modules, the run module which encapsulates the more simple and more time critical requirements which can be performed during a normal link operation, and the shutdown module which contains the more advanced requirements.

4.1 The Run Module

The run modules main tasks is to continuously extract and record data for future use in the shutdown module and to detect critical faults on the video link. The following requirements describes the content in the run module of the SIVIC concept.

The SIVIC application shall be able to meet the following requirements:

- **REQ 1:** Read and write register data to both the serializer and deserializer side of the communication.
- **REQ 2:** Work of different test rigs.
- **REQ 3:** Continuously record internal and external parameters for diagnostic purposes.

- **REQ 4:** Structure and store the recorded internal and external data.
- **REQ 5:** Detect both internal and external issues and critical faults by looking the recorded parameter data.
- **REQ 6:** Take actions to temporarily remedy link defects.

4.2 The Shutdown Module

The shutdown module's main tasks is to use the data recorded by the run module to make statistical analysis and find trends of video link performance. The following requirements describes the content in the shutdown module of the SIVIC concept.

The SIVIC application shall be able to meet the following requirements:

- **REQ 7:** Make statistical analysis of recorded data.
- **REQ 8:** Use statistical analysis for detecting trends.
- **REQ 9:** Initiate built in test sequences during power down and compare the results with the previous test sequences.
- **REQ 10:** Determine thresholds for link quality.
- **REQ 11:** Warn when the link quality is no longer trustworthy.

5
Proof of Concept

This chapter describes the implementation of the SIVIC application developed to meet the requirements of the SIVIC Concept. This chapter represents the development phase of the design science research methodology mentioned in section 2.1.

5.1 The Run Module Implementation

This section covers the implementation of the SIVIC application to fulfill the requirements of the run module.

5.1.1 Read and write register data over UART

Requirement 1 in the SIVIC concept in section 4.1 states that the SIVIC application shall be able to read and write data to both the serializer and deserializer side of the communication. This requires communication between a computer and the serializer-deserializer circuits. The included control channel in GMSL, based on UART allows this to take place. By using the duplex UART protocol, it is possible to read and write messages to the registers of both serializer and deserializer circuits.

This functionality is the the base of the SIVIC concept. It enables both data collection for future analysis and immediate fault detection. Figure 5.1 presents how the data packets are structured while writing and reading data from registers using UART on GMSL. Each byte in the data frame is structured as a UART data frame according to Figure 3.4.

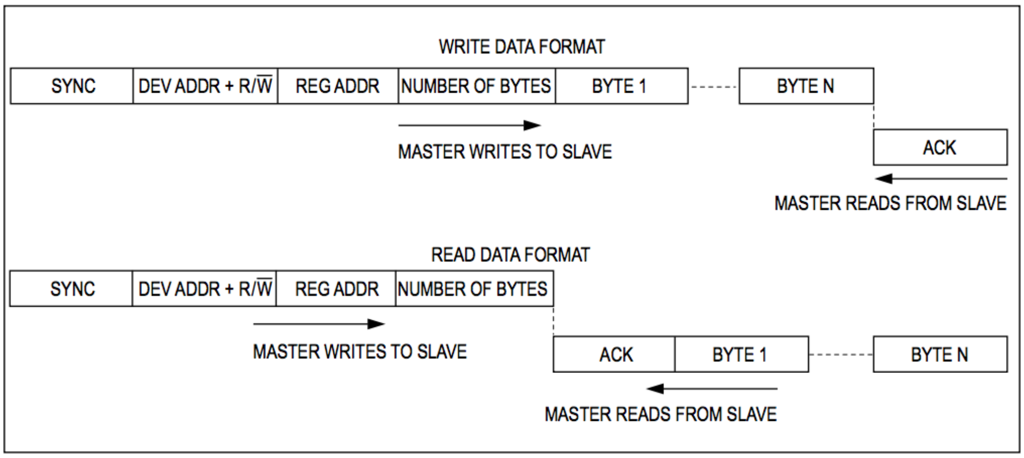

Figure 5.1: GMSL-UART protocol from [8].

5.1.1.1 Write register

The data package for writing to a register contains 6 parts:
1. Sync byte
2. Device address byte
3. Register address byte
4. Number of bytes
5. Data bytes
6. Acknowledgement byte

The sync byte makes sure that both circuits are ready to send information between each other. The device address byte contains the address of the device (whether it is the serializer or deserializer) on which the operations should be carried out on. The register address byte contains the address of the start register to which the data should be written. Number of bytes states how many bytes of data the message contains. Data bytes contain the actual data to be written. If there are more than one data byte, the successive bytes will be written to the next register after the register with register address byte. Finally, the acknowledgement byte is sent from the device where the register writing have been carried out on alerting that the process has been successful.

5.1.1.2 Read register

The data package for reading a register contains the same six parts as the write data package, but in a different order. Figure 5.1 displays that the first four parts of the message is the same as when writing data. The difference here is that the acknowledgment byte comes directly after the device on which the operations are being carried out on, has received the number of bytes of data that is going to be read. The acknowledgement byte alerts that the read request have been understood and then the data bytes follow in the order. If there has been a request to read more than one data byte, the readout starts at the register specified in the register address byte and then reads from the next register in order.

5.1.2 Usage profiles

Requirement 2 in section 4.1 in the SIVIC concept states that the SIVIC application shall be able to work on different test rigs. To meet this requirement the SIVIC application will have a usage profile for each test rig containing the test rig specific settings and parameter information. These usage profiles is used in the other parts of the SIVIC application to specify which parameters can be recorded and what other information is available to process. They are also used during every active session of the video links to verify that everything is up and running correctly.

Parameter information

Each parameter in the usage profile will have an information struct related to it. Table 5.1 describes how this information struct could look like for two different parameters. This struct contains information about its whereabouts such as the type of content it is based on and to which usage profile it belongs to. There are mainly two different kind of data types, counter and toggle. The toggle data type consists of one bit which can be either 1 or 0. This is to flag when an unexpected event happens, such as too many packets have been re-transmitted, hence the max re-transmission flag is set to 1. The counter data type on the other hand contains multiple bits to include a larger set of numbers, often one byte. For the example in table 5.1 the decoding error parameter is a counter which calculates the amount of detected bit errors since the last time the register was read. This can be a value between 0 and 255.

Table 5.1: Structure of parameter information structs.

Information	Bit error	Max retransmission
Parameter name:	Bit error	Max retransmission
Source:	Deserializer	Serializer
Register:	0x0D	0x16
Type:	Counter	Toggle
Bytes:	1	1
Bit:	N/A	6
Default:	0	0
Min:	0	0
Max:	255	1
Data type:	Integer	Integer
Usage profile:	PAC	PAC

5.1.3 The implementation of signal data recording

Requirement 3 and 4 in section 4.1 in the SIVIC concept states that the SIVIC application shall be able to continuously record internal and external parameters for diagnostic purposes, structure the data and then store it. Since the parameter contents is going to be revisited during future analysis and evaluation, an external data storage implementation is used. The commonly used comma separated value (CSV) file type is chosen as it is a good way to store data in a structured way.

The parameter recording function operates continuously when the video link is active. It receives a usage profile as explained in section 5.1.2 as an argument which contains the parameters that are possible to read for out as well as containing information describing each of these parameters. The parameters are then monitored one by one, and when the contents of a parameter has been recorded, the module shifts its focus to the subsequent parameter listed in the usage profile. When the last parameter stated in the usage profile has been reached, the module starts over again with the first one and continues until the video link is turned off.

To start the recording of a parameter, the parameter recording function loads a struct as described in table 5.1 containing the associated information essential to building a structured data log. By using the address and device information in the struct, the module is able to read out the content of the parameter according to the method explained in section 5.1.1.2. This content alone is not enough to make future analysis since there are some other system status values which also should be incorporated to the recorded data. These are time stamp, the active session counter, timeout flag and fault detection flag. The time stamp is just a snapshot of the clock at the exact moment from when the content of the parameter was read out. This can be used to sort the data into specific increments with a very small interval. The active session counter is crucial for the future statistical analysis. Every time the video link is initiated, this counter is incremented by one. This gives the possibility to sort the data into intervals from each use of the video link. This is useful since the environment and disturbances can vary drastically depending on the whereabouts of the car, which can change for each active session of the video link. The timeout flag is activated by the fault detection module described in section 5.1.4 and is activated when the register readout according to section 5.1.1.2 has failed. The fault detection flag is also activated by the fault detection module. This is activated when the parameter content is above a certain threshold indicating video link failure.

When the content of the parameter has been read, and the surrounding values have been amassed, the recording module opens an external CSV file and appends this information below all the previous information stored in this file. If there are no previous recorded data, then the module creates a new file. Each parameter's content is logged in a separate file and table 5.2 is a sample of the recorded data of decoding errors. This presents the structure of how the contents are stored.

Table 5.2: Data structure for storing internal data.

Content	**Time**	**Active session counter**	**Timeout flag**	**Fault detected**
0	Thu Apr 20 14:02:49 2017	1	0	0
0	Thu Apr 20 14:02:50 2017	1	0	0
23	Thu Apr 20 16:35:20 2017	2	0	0
20	Thu Apr 20 16:35:20 2017	2	0	0
255	Fri Apr 21 10:20:10 2017	10	0	1
255	Fri Apr 21 10:20:11 2017	10	0	1

External environment data can also be recorded by the choice of the user. External data can contain ambient temperature, barometric pressure, battery voltage and humidity. All of these parameters are recorded at the same time as the internal parameters. The environmental data is appended to the original internal data structure depicted in table 5.2. Table 5.3 describes the fields which the original table will be expanded with.

Table 5.3: Data structure for storing external data.

Ambient temperature	Barometric pressure	Battery voltage	Humidity
6 °C	1020 mb	12 V	68 %
6 °C	1020 mb	12V	68 %
6 °C	1020 mb	12 V	68 %
6 °C	1020 mb	12 V	68 %
15 °C	1005 mb	12 V	80 %
15 °C	1005 mb	12 V	80 %

5.1.4 Fault detection

Requirements 5 in section 4.1 in the SIVIC concept states that the SIVIC application shall be able to detect both internal and external issues and critical faults by looking at the recorded parameter data. This is to detect video link failure during link operation and thus opening up the possibility to alarm the user if the video link is not reliable any more. To detect failure immediately, the parameters are sent through a fault detection process directly after they have been read from their register according to section 5.1.1.2, even before they are recorded and structured according to section 5.1.3.

Depending on what test rig is in use, the fault detection process must be aware of the parameters to evaluate. Here the usage profile described in section 5.1.2 containing parameter information is crucial for proper functionality. The parameters differ significantly, some indicate total video link failure while some indicate that something may be wrong but the video link is still working properly. The faults can be divided into three different communication levels. Each explained in it's own section below.

Faults in the physical layer of the OSI model

The two most significant parameters for detection of critical errors contain information about cable faults and loss of video link lock. Both are of the data type toggle described in section 5.1.2, which means they are 0 during normal operation and 1 when activated.

The cable fault parameter is activated if the cable of the video link in some way have been disconnected or shorted. If this would happen, then the video link would not work at all and can thus be seen as a critical fault. The video link lock parameter is activated if the serializer and deserializer looses their communication lock due to too many errors in the communication. Video link lock means that both the serializer and deserializer have a synchronized clock, enabling data transfer over the video link. If there is no lock, no data can be transferred which can be seen as a critical fault.

Another parameter which contain more information about the general status of the video link is the eye width monitor. The signal eye diagram is a tool for analyzing the overall status of the video link. The eye width parameter is of the data type counter described in section 5.1.2, and the parameter counter contains information about how wide the eye diagram is compared to it being fully open. This is described in percentage. Minor faults on the video link could be discovered through this parameter. If the video link is working properly and the eye width is narrower than usual, then this could be an indication that something went wrong. Also if the eye diagram is very narrow, then critical faults have occurred.

Faults in the video link layer of the OSI model

There are also some parameters containing information which could be useful for detection of video link faults on levels above the physical layer. In the video link layer there are mainly three different parameters which contain information about the video link status. The first parameter contains information about the number of bit errors which have been detected on the video link. This is of the data type counter and is basically just calculating the amount of bit errors since the last time it has been read. This parameter can be used to detect when something may be wrong but the video link is still running. Also, if the number of bit errors detected is greater than a certain threshold then it can be used to detect critical faults.

The second parameter detects when there has been a high number of attempts to resend a symbol without succeeding. This is of the data type toggle and is activated if it detect high number of attempts. This can be used as an indication of deteriorating link quality. The third parameter detects when there are large deviations of the amount of pixels per line received over the video link. This basically means the data transmission is inconsistent, therefore, can be used to indicate deteriorating link quality. This is also of the data type toggle.

Faults in the transport layer of the OSI model

Issues can also be detected on even a higher levels of communication. If there are many problems on the data transmission then there could be problems reading and writing data to registers, basically because all messages contain bit errors. This is often a problem of time, when it takes too long time to write or read the registers. If there are too many errors, then you can not deliver full write or read requests, which means the video link is not working.

5.1.5 Remedy link defects

Requirement 6 in section 4.1 in the SIVIC concept states that the SIVIC application shall be able to take actions to temporarily remedy link defects. This is for enabling video link usage until it has been repaired or replaced. Using the built in equalization and preemphasis feature explained in section 3.5, the signal quality can be improved by increasing the levels until the issues are decreased and the communication is working on a good level again.

5.2 The Shutdown Module Implementation

This section covers the implementation of the SIVIC application to fulfill the requirements of the shutdown module.

5.2.1 Calculations for analysis

Requirement 7 in section 4.2 in the SIVIC concept states that the SIVIC application shall be able to make statistical analysis of the recorded data. To carry out the trend analysis it is important to detect the different disturbances affecting the video links during each active session. Section 3.9 explains two kinds of major disturbances; continuous and transient disturbances. For detecting these different kinds of disturbances, the data need to be analyzed in different ways.

Continuous disturbances are constantly affecting the video link which is being exposed to it. If the disturbance level are at approximately the same level during an active session, then the amount of bit errors generated and the measured eye width will be about the same level during the active session. By using the mean value equation (5.1) on the level of bit errors and the eye width measurement, these two mean values can form a representation for severity of the continuous disturbances during the active session. The value $\bar{x}$ is the calculated mean value, n is the size of the data set and x_i is each individual sample.

$$\bar{x} = \frac{\sum_i x_i}{n} \tag{5.1}$$

Transient errors are a bit harder to detect, they could appear periodically or sporadically. One thing high amplitude transient disturbances have in common is that they can cause severe deviations in both bit errors and eye width measurement during the transient pulse. So the parameter values may vary significantly during a pulse and during normal operation. A way to measure if these transient disturbances have occurred during an active session of the video link is by calculating the standard deviation according to equation (5.2) which is a measurement of the spread of the recorded parameter data. The parameter s is the calculated standard deviation, $\bar{x}$ is the mean value of the data set, x_i is each individual sample and n is the size of the data set.

$$s = \sqrt{\frac{\sum_i (x_i - \bar{x})^2}{n-1}} \tag{5.2}$$

If the video link would be exposed to both continuous and transient errors during an active session, then both the calculated means and standard deviation can be used together to detect this.

As external factors such as temperature and humidity could affect the performance of the video link, the mean and standard deviation can also be calculated for these during an active session enabling future analysis of correlation. For every active video link session, the amount of detected critical faults and failed register read/ write commands are summed up for comparison to other active video link sessions.

5.2.2 Trend analysis

Requirement 8 in section 4.2 in the SIVIC concept states that the SIVIC application shall be able use statistical analysis for trend detection. Here the mean values, standard deviations and sum of faults from section 5.2.1 are used and compared to the corresponding values of previous active video link sessions.

Trends are often a change over time but they can come in many different shapes and sizes. To cover rapidly developed trends, trends developed over a long time and trends which is somewhere in between these two, a method named frame analysis was developed. A frame is basically the combined data from multiple adjacent active video link sessions. This data is then analyzed using the mean value equation (5.1) and standard deviation equation (5.2) to get the mean and standard deviation of the total frame. To cover these different trends, frames of different sizes are used. To detect rapidly developed trends, a short frame consisting of two or three active video link sessions are compared to the previous frame of the same size. To detect fairly developing trends, a medium frame consisting of five to seven active video link sessions are compared to the previous frame with the same size. To detect trends developed over a long time, a long frame consisting of 10 to 15 active video links sessions are compared to the previous frame with the same size. Figure 5.2 present how the frames are built and differ from each other.

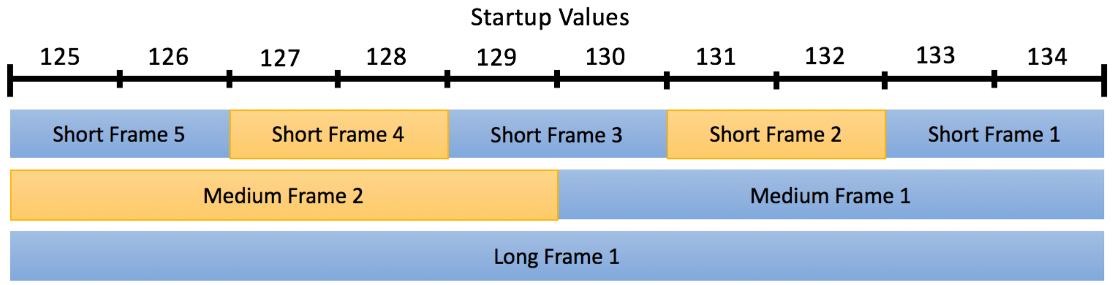

Figure 5.2: Data Frame description.

5.2.3 PRBS test and analysis

Requirement 9 in section 4.2 in the SIVIC concept states that the SIVIC application shall be able to initiate built in test sequences during power down and compare the result with previous test sequences. This is a good way to examine the status of the video link after detecting deteriorating behaviour in the trend analysis. The GMSL serializer and deserializer circuits have a built in pseudo random bit sequence (PRBS) function for performing video link tests. When activated, it switches from the regular video transmission mode and the serializer, instead sends a pseudo random bit stream which the deserializer analyzes and calculates the amount of detected bit errors.

To fulfill this requirement, the SIVIC application upon detecting an inactive video link, starts this PRBS function for a specified duration by the user. Afterwards it reads the register with the amount of detected bit errors and stores it in a CSV file. The amount of detected bit errors during this test is then compared to the previous PRBS test results. If there are any detected bit errors during this PRBS test, then the video link is probably broken.

5.2.4 Thresholds for link quality

Requirement 10 in section 4.2 in the SIVIC concept states that the SIVIC application must be able to determine thresholds for link quality. This comprises thresholds for detected bit errors and the width of the eye diagram to detect critical faults. The user of the SIVIC application shall easily be able to change these thresholds. Before every active video link session, the SIVIC application will print the current thresholds and then the user can decide if they want to change the thresholds by writing the new threshold values in the console. If the user does not want to change the existing threshold, then he/she can input "no" in the console which will keep the current threshold values and continue the SIVIC application.

5.2.5 Warn when no longer dependable video link

Requirement 11 in section 4.2 in the SIVIC concept states that the SIVIC application shall be able to warn when the link quality is no longer dependable. The PRBS test is a raw test of the serial link without any other factors weighing in. If the PRBS test detects errors, then the SIVIC application will print a message stating that the serial link is not reliable any more and should be inspected.

6 Evaluation

In this chapter the evaluation step of the design science research was performed. Here the artifact developed during the development stage will be analyzed to see if it solves the research problem of the thesis.

6.1 Test Environment

This section discusses the hardware specifics of the test rigs as well as the disturbance sources.

6.1.1 Test rig: parking assist camera

The main test rig used in this thesis was previously built by Diadrom as a proof of concept for a parking assist camera. It contains a camera with a GMSL serializer of model MAX96705 [8] and an ECU with a GMSL deserializer of model MAX9288 [22]. The ECU is also connected to an image grabber enabling the computer to receive the video from the camera. Figure 6.1 shows an overview of the parking assist camera system and how the parts are interconnected. The computer containing the SIVIC application is connected to the MAX9288 circuit by using a Universal Serial Bus (USB) cable for UART called FTDI. This cable is used to send read and write commands over the control channel of the GMSL link. The computer is also connected to the image grabber with a regular USB cable in order to send the video from the camera to the computer. The serializer and deserializer is connected via a STP cable which in this test rig is the high speed automotive video link. Figure 6.2 depicts the parking assist camera test rig.

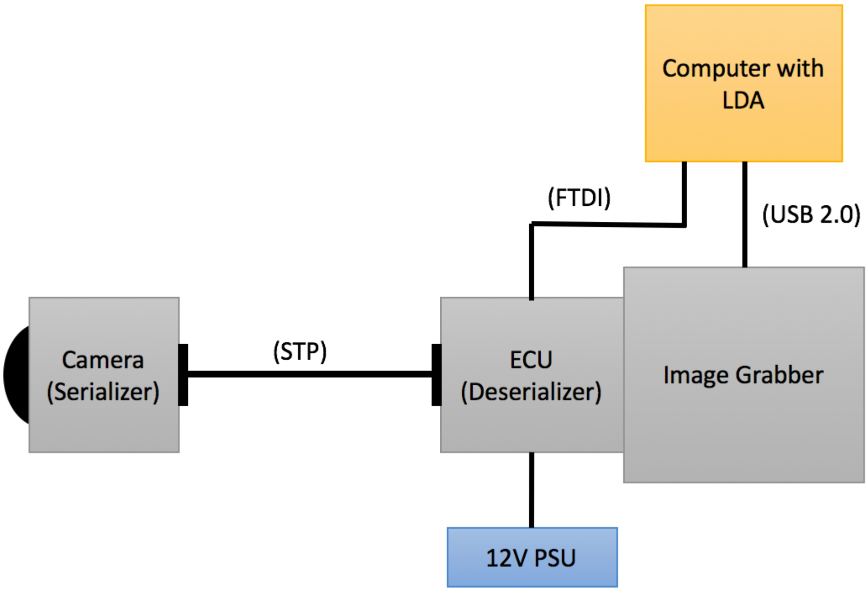

Figure 6.1: Overview of parking assist camera hardware.

This test rig allows fault detection with the following parameters: bit error detection, pixel per line error, line fault detection, loss of video link lock and max retransmission errors. The qualities of these parameters are explained below.

Bit error detection

This parameter stores the amount of detected bit errors on the serial link. Bit errors are detected by using the 8b/10b encoding/decoding and the 1 bit parity.

Pixel per line error

This parameter raises a flag if the amount of pixels per line in a data enable high period deviate more than ±4 pixels compared to the previous pixel count. Enabling this flag suggests that there may be disturbances on the serial link.

Line fault detection

This parameter contains information about the status of the physical serial link. It detects if the cable in some way is disconnected or shorted.

Loss of video link lock

This parameter raises a flag if the deserializer looses the video link lock which means that the serializer and deserializer is no longer in sync. The system looses video link lock when too many bit errors are detected.

Max retransmission errors

This parameter raises a flag if the serializer tries to resend a packet eight times without succeeding over the UART control channel. This is an indication that the system may be exposed to disturbances.

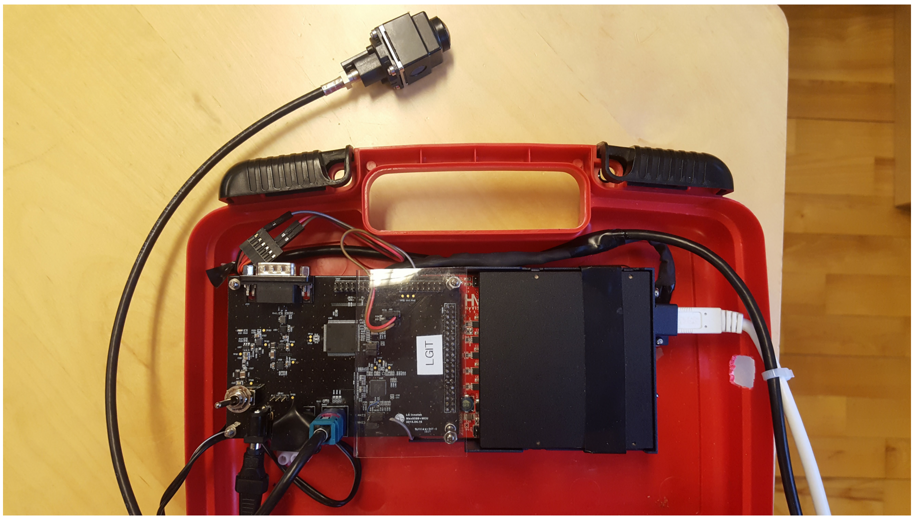

Figure 6.2: Parking assist camera test rig.

6.1.2 Test rig: evaluation boards

The secondary rig used in this thesis is built by two so called evaluation boards used for GMSL system development. It contains a GMSL serializer MAX9291 [23] and a deserializer MAX96706 [24]. These circuits are updated versions of the circuits used in the parking assist camera test rig and have some added functionality such as an eye width monitor which enables analysis of the signal quality in the physical layer. The serializer receives video data from a connected HDMI cable from a video source. The deserializer is connected to the computer containing the SIVIC application with a FPDI cable. This test rig lacks an image grabber and because of this, the computer with the SIVIC application can not access the image from the video source. The serializer and deserializer is connected via a STP cable which in this test rig is the high speed automotive video link. Figure 6.4 depicts the evaluation boards test rig.

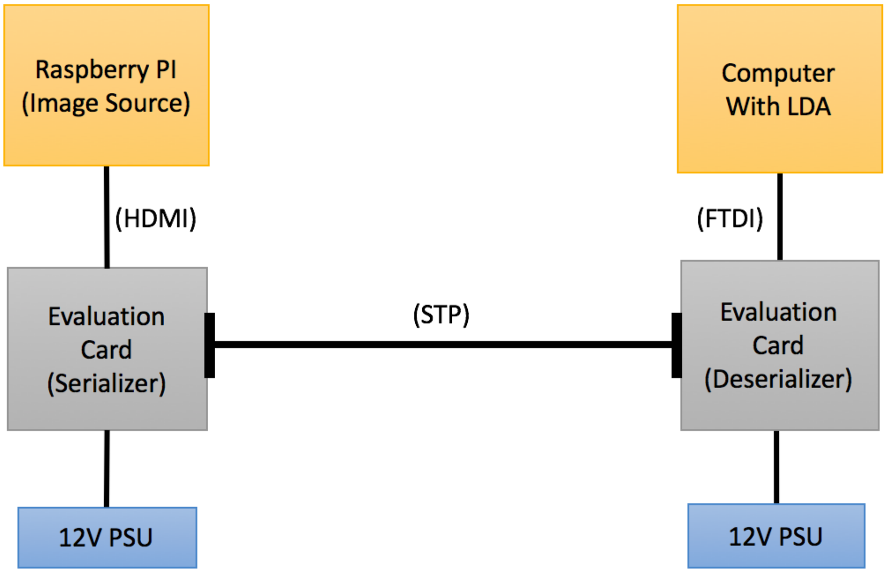

Figure 6.3: Overview of evaluation board hardware.

This test rig allows fault detection with the following parameters: bit error detection, eye width monitor, line fault detection, loss of video link lock and max retransmission errors. The qualities of these parameters are explained below.

The qualities of the parameters: bit error detection, line fault detection, loss of video link lock and max re-transmission are all described in section 6.1.1.

Eye Width Monitor

This parameter contains information about the measured eye opening of the signal. The opening is represented in percentage and can be used to evaluate the quality of the signal. If the eye opening is narrow, then it is an indication that the system may be exposed to disturbances.

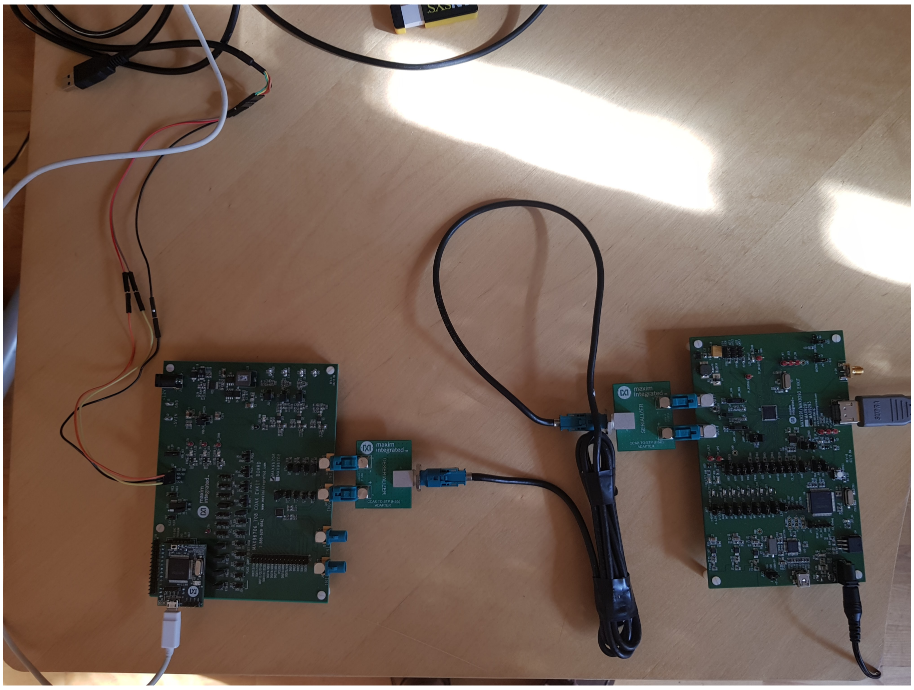

Figure 6.4: Evaluation Boards Test Rig.

6.1.3 GMSL built in error generation

Some versions of the GMSL serializer and deserializer cards have an included error generation function for performing tests on the video link. This enables the developer to inflict bit errors of different magnitudes and lengths to the regular data transmission. This is a useful feature for development of diagnostic software since it enables disturbance injection on the video link without the need for external hardware.

The serializer circuit MAX96705 [8] in the camera of the parking assist camera contains this feature. Table 6.1 contains the settings for the error generating function inside MAX96705. All of these settings can be combined which makes it possible to generate $4 * 4 * 4 = 64$ types of errors. The error generation feature is activated by writing a message to a register in MAX96705, containing the error generation settings combined with information used for enabling the feature. To deactivate the feature, a message to disable the feature is sent to the same register.

Table 6.1: Built in error generation functionality in MAX96705.

Error generation rate				
Error frequenzy	2560 bits	40,960 bits	655,360 bits	10,485,760 bits
Error generation type				
Error size	Single-bit	2 8b/10b symb	3 8b/10b symb	4 8b/10b symb
Error generation count				
Error quantity	Continous	16 errors	128 errors	1024 errors
Periodic Error Generation enable				
Error mode	Random generation of errors		Periodical generation of errors	

6.1.4 Transient disturbance injection at RISE

Research Institute of Sweden (RISE) [25] contributed to this thesis by supplying both hardware and a lab environment to carry out tests for evaluation of the SIVIC application. The hardware consists of a transient immunity test generator of model RCB200N1 [26] and a test fixture for inducing these disturbances on the test rigs [9]. This hardware is able to conduct tests of unintended transient disturbances caused by coupling of two adjacent wires according to the standard RI 130 [9]. These disturbances are emitted by components such as electrical motors containing electrical switching.

The standard includes two different transient pulses A2-1 and A2-2, which can be generated in by either a pseudo random timing sequence or pseudo random bursts. These generation patterns and pulses are shown in appendix A. The different pulses combined with the generation patterns make it possible to generate four different kinds of transient disturbances. These are described in the standard [9].

Figure 6.5 presents how the transient generating hardware is connected to the evaluation board test rig without including the details. It is connected to the parking assist camera test rig in the same way, over the STP cable. A copper wire connected to the transient generator passes through the test fixture and then into the test generator again for termination of the signal. Inside of the test fixture, this copper wire is positioned right next to the STP cable of the test rig in order to induce the disturbances.

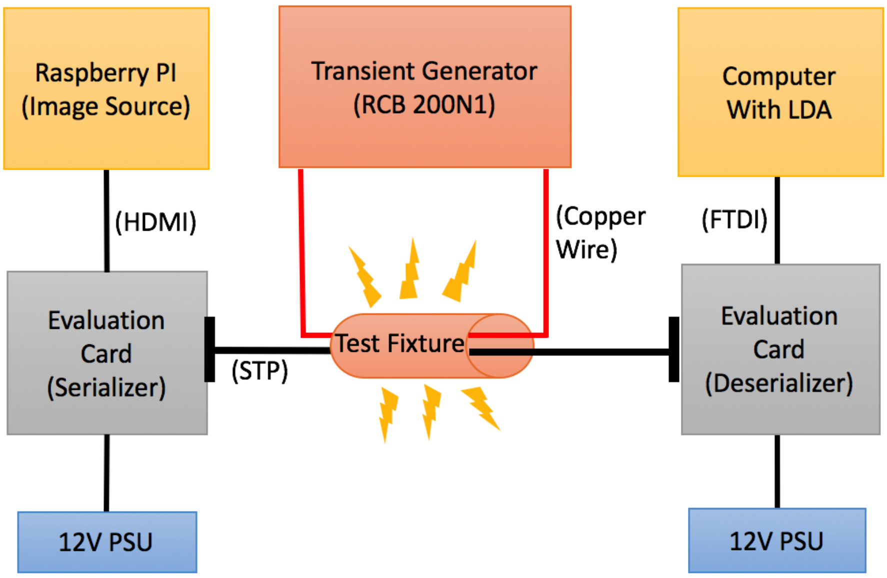

Figure 6.5: Overview of evaluation board hardware with injected disturbances.

RI 130 have a rigorous specification for how these tests should be set up and performed. All the steps except one is followed. The standard suggests that the shield of a STP cable should be broken and the cable should be untwisted in order to only inject the transient disturbance on one of the internal cables. Unfortunately the video link immediately looses its functionality when the shield is broken. Instead the transient disturbances were injected on a unbroken STP cable.

6.2 Tests for Evaluation

This section includes test description for tests performed in order to evaluate the quality of the implementation of the SIVIC application.

6.2.1 Read and write register data over UART

To evaluate if the implementation of the SIVIC application is able to fulfill requirement 1 of the SIVIC concept, a test is done where read and write commands are sent to registers in both the serializer and deserializer. The read register functionality is tested by reading the content from two different registers, one in the serializer and one in the deserializer. The result of these readouts are then compared to the contents specified in the data sheets to verify a correct readout. The read register test is passed if both register have been successfully read and the contents are verified to be correct.

The write register functionality is tested by first reading contents from two different registers, one in the serializer and one in the deserializer. The reason for reading the content of these registers is to verify that they do not already contain the same message which is going to be written to them. After the verification of the contents, a message will be written to these registers. They will then be readout once again and the contents is compared to the write message to verify if it has been written correctly. The write register test is passed if the message were successfully written to the registers.

6.2.2 Usage profiles

To evaluate if the implementation of the SIVIC application is able to fulfill requirement 2 of the SIVIC concept, a test is conducted where the user switches between the test rigs and writes only one command to the SIVIC application to specify what test rig is in use. The test is passed if the SIVIC application is able to change all the relevant settings for setting up the different test rigs by only receiving one command by the user.

6.2.3 Recording of parameter data during an active video link

To evaluate if the implementation of the SIVIC application is able to fulfill requirements 3 and 4 of the SIVIC concept, a test where the video link is active for 30 seconds and all fault detection parameters is recorded and stored. The test is passed if the SIVIC application is able to record and store all fault detection parameters associated to the test rig in use according to the structure described in section 5.1.3.

6.2.4 Fault detection

To evaluate if the implementation of the SIVIC application is able to fullfill requirement 5 of the SIVIC concept, a test where the worst disturbance possible to generate with the built in error generation feature explained in 6.1.3 is injected in the parking assist camera test rig. This disturbance generates continuous errors of four 8b/10b symbols every 2560 bit periodically for 30 seconds. The test is passed if the SIVIC application is able to use the fault detection implementation to detect the critical faults during this test.

6.2.5 Effect of transient disturbances using RISE equipment

To evaluate how the disturbances generated by the equipment described in section 6.1.4 affect the GMSL link and also see how the fault detection implementation works with real life disturbances, a series of tests were performed. These are not tests to fullfill any requirements of the SIVIC concept. It is instead used to evaluate how the test rigs are affected by the injected real life disturbances by using the fault detection implementation. There will be four tests where the different transient disturbances explained in 6.1.4 will be injected on the video link during an active video link session. There will also be a test where the shield of the video link is broken during an active video link session.

The test 1 will expose the video link with pulse A2-1 in a pseudo random timing sequence during an active session. Test 2 will also use the A2-1 pulse but change the sequence to pseudo random bursts. Similarly, test 3 will use the A2-2 pulse in a pseudo random timing sequence and test 4 will use the pulse A2-2 with pseudo random bursts. These four tests combined cover the range of the transient inducing equipment. Test 5 would check how the SIVIC application handles a broken shield of the video link cable during an active video link session. Since the goal of these tests are not to fullfill any requirements, there is no pass or fail.

6.2.6 Temporarily remedy link defects

To evaluate if the implementation of the SIVIC application is able to fullfill requirement 6 of the SIVIC concept, a test where the test rigs are exposed to continuous errors of two 8b/10b symbols every 10,485,760 bit periodical by using the built in error generation functionality. The test is passed if the SIVIC application is able to use equalization and pre-emphasis to make the serial link more robust and decreases the amount of bit errors.

6.2.7 Calculation for analysis

To evaluate if the implementation of the SIVIC application is able to fullfill requirement 7 of the SIVIC concept, a series of tests will be performed to analyze if the statistical analysis implementation is able to detect and differentiate between continuous and transient disturbances. The Built in error generation feature described in section 6.1.3 will be used to generate the disturbances. To test the continuous disturbance detection, an error sequence that has the error frequency four 8b/10b symbol errors in every 10,485,760 bit, generated continuously in a periodical pattern will be introduced on the video link.

To test the transient disturbance detection, an error sequence that forms an transient disturbance is built by inducing two 8b/10b symbols every 2560 bit for

16 times. This is one transient pulse and will be inflicted on the video link two times during a start up. To tests if the SIVIC application is able to detect if the system is exposed to both continuous and transient disturbances at the same time, the first and second test will be combined into a third test. The tests are passed if it is possible to distinguish transient and continuous disturbances by using the statistical methods described in section 5.2.1.

6.2.8 Trend analysis

To evaluate if the implementation of the SIVIC application is able to fullfill requirement 8 of the SIVIC concept, a series of tests will be performed to analyze if the trend analysis implementation is able to detect fast, slow and medium changing trends. The built in error generation feature described in 6.1.3 is used to generate the disturbances.

To test rapidly developing tests, a test sequence divided into two parts is introduced. The first part includes two active video link sessions in a consecutive order, forming a short frame with a disturbance of 1 bit error every 10,485,760 bit continuously over both sessions. The second part also contains two active video link sessions in a consecutive order but they all are exposed to a slightly worse disturbance. They are all exposed to a disturbance of four 8b/10b symbols in every 10,485,760 bit continuously in a periodical pattern.

To test fairly developing trends, a test sequence almost exactly like the test for fast trends are introduced. The only difference is that, this test uses five consecutive active video link sessions in a row instead of two, forming a medium frame.

To test trends developed over a long time, a test sequence almost exactly like the test for fairly developing trends are introduced. the only difference is that this test uses ten consecutive active video link sessions in a row instead of five, forming a long frame. The tests are passed if the SIVIC application is able to detect an increase in disturbances over the frames.

6.2.9 PRBS test and analysis

To evaluate if the implementation of the SIVIC application is able to fullfill requirement 9 of the SIVIC concept, a test where transient disturbances will be injected on the high speed video link during a PRBS test. The transient disturbance will be generated by the equipment in the lab at RISE and will consist of a A2-1 pulse in pseudo random burst sequence. The test is passed if the SIVIC application succeeds with the following requirements. It is able to launch a PRBS test after an active video link session where bit errors have been detected. It is able detect bit errors during this PRBS test from the induced disturbance, and it is able to compare the amount of detected bit errors with the result from previous PRBS tests.

6.2.10 Thresholds for link quality

To evaluate if the implementation of the SIVIC application is able to fullfill requirement 10 of the SIVIC concept, a test is conducted where the user of the SIVIC application is requested to change the threshold for fault detection of bit error detection and eye width monitor. The test is passed if the change of thresholds are successful.

6.2.11 Warn when no longer dependable video link

To evaluate if the implementation of the SIVIC application is able to fullfill requirement 11 of the SIVIC concept, a test where the test rigs are exposed to continuous errors of two 8b/10b symbols in every 40,960 bits periodically by using the built in error generation functionality is performed. An error of this quantity will disable the functionality of the video link and make it non dependable. The test is passed if the SIVIC application is able to detect the critical faults and print a message to the user stating that the link is exposed to high disturbances, hence unreliable.

6.3 Test Results

This section covers the results of the tests for each part of the SIVIC application described in section 6.2. The result of the more critical parts of the SIVIC application will be described in detail while the results of the remaining parts will is show only in table 6.2.

Table 6.2: Final result of all tests.

Requirement	**Test**	**Passing acceptance criteria**
Req: 1	Read and write register data over UART	Yes
Req: 2	Usage profiles	Yes
Req: 3 & 4	Recording of parameter data during and active video link	Yes
Req: 5	Fault detection	Yes
Req: 6	Temporarily remedy link defects	No
Req: 7	Calculation for analysis	Yes
Req: 8	Trend analysis	Yes
Req: 9	PRBS test and analysis	Yes
Req: 10	Thresholds for link quality	Yes
Req: 11	Warn when no longer dependable video link	Yes

Recording of parameter data during an active video link

The SIVIC application was able to record every fault detection parameter in each test rig about 3.3 times in every second resulting in about 100 samples over the 30 second test. The SIVIC application was able to store everything according to the structure described in section 5.1.3 without mixing the data meaning it has passed the test.

Fault detection

Table 6.3 comprises the results of the fault detection test. As the SIVIC application is able to read every parameter content about 3.3 times in every second. It means for a 30 second test, the SIVIC application is able to read the contents about 100 times. With a disturbance of this level, the video link is practically useless. The Fault detection have registered 99 critical errors of a high number of detected bit errors as well as 91 failed register read/ write commands. The fault detection implementation is able to detect the critical faults, therefore, the test is passed.

Table 6.3: Result of Fault detection test using built in disturbance Feature.

Fault detection parameters	Built in error generation test
Detected bit errors	99
Detected loss of video link lock	2
Detected max retransmissions	0
Detected failed read or write commands	91

Effect of transient disturbances using rise equipment

Table 6.4 and 6.5 includes the results of the transient tests induced by the equipment at RISE. By looking at the results, a conclusion can be drawn that the evaluation board test rig is much more susceptible for external disturbances while the parking assist camera is practically not affected at all. Both detect a line fault error during the broken cable shield test, which resulted in a broken communication.

Table 6.4: Effect of the induced transient errors on the evaluation board test rig by equipment at RISE.

Fault detection parameters	Test 1	Test 2	Test 3	Test 4	Test 5
Detected bit errors	3	2	13	12	0
Detected narrow eye width	43	32	71	73	0
Detected loss of video link lock	44	29	67	64	0
Detected max re transmissions	2	1	3	3	0
Detected failed read or write commands	0	0	3	2	0
Detection of line faults	0	0	0	0	1

Table 6.5: Effect of the induced transient errors on parking assist camera test rig by equipment at RISE.

Fault detection parameters	Test 1	Test 2	Test 3	Test 4	Test 5
Detected bit errors	0	0	0	0	0
Detected loss of video link lock	0	0	0	0	0
Detected max retransmissions	0	0	0	0	0
Detected pixel per line deviations	1	0	0	0	
Detected failed read or write commands	3	0	0	0	0
Detection of line faults	0	0	0	0	1

Calculations for analysis

The table 6.6 includes the results from the tests of the calculation for analysis implementation. From the continuous disturbance test, the mean value of detected bit errors is quite high but the standard deviation is low. However, from the transient test, the mean value of detected bit errors is low but the standard deviation is quite high. From the combined continuous and transient disturbance test, both mean value and standard deviation are quite high suggesting that the system is exposed to both transient and continuous disturbances if compared to the results from the previous tests. It is possible to distinguish between the two different disturbances by using only mean values and standard deviation, therefore, the tests are passed.

Table 6.6: Result of calculation for analysis tests.

Statistic results of bit error detection	**Continuous disturbance test**	**Transient disturbance test**	**Combined disturbances test**
Mean Value	44.40	5.09	55.68
Standard Deviation	3.70	35.28	29.33

Trend analysis

Table 6.7 includes the result from tests for the trend analysis implementation of the SIVIC application. The SIVIC application is able to detect an increase of about 80% bit errors over the frames for all of the tests. The SIVIC application is thus able to detect fast, medium, slow changing trends by corresponding frames respectively. It has fulfilled the requirement and thus passed the test.

Table 6.7: Result of trend analysis tests.

Fault detection parameters	**Detected difference in bit errors over two frames**
Fast trend test	84 % more bit errors in second frame
Medium trend test	80 % more bit errors in second frame
Long trend test	79.3 % more bit errors in second frame

Temporarily remedy link defects

The SIVIC application was not able to change the Equalization and Preemphasis levels and thus it has not passed the test.

7
Discussion

The goal of this thesis is to explore if it is possible to build a diagnosing software for analysis of high speed data links in future vehicles. By following the research methodology based on iterative steps called design science research as an approach, a software named Simons video interference check was developed. I proved that an diagnostic application with this functionality is in fact possible to build, and observed that the experience of using this iterative methodology is very positive. In the light of the knowledge gained during the preparation of this thesis, it was possible to go a few steps back in terms of design and development of the software thereby resulting in a better final product.

The final version of the SIVIC application is robust and is able to perform diagnostics of the data link. By storing the parameter data in separate CSV files for each different parameter, mix up of data was absolutely impossible. This way of storing the data also created a structure which was easy to use and easy to find information from. The thesis was limited to only use data from the fault detection parameters which proved to be enough to analyze the link quality. However, the eye width monitor is the only parameter supplying information about the link quality directly from the raw data signal. All other parameters relied on the underlying features implemented by Maxim Integrated such as detection of too many re-transmitted symbols and detected bit errors. The eye width monitor allows the user of the GMSL link to make one's own analysis while the other parameter forces one to trust that the system is able to detect all errors and issues on the data link.

Unfortunately, I only gained access to the evaluation board test rig with the eye width monitor feature in the final weeks of the thesis. All analysis were developed around the other fault detection parameters implying that the full potential of the eye width monitor is not fully used. Although the analysis still fullfill its requirements without any problems. The choice of the frame analysis for the trend detection is good and it is able to detect different levels of trends. The choice of mean value and standard deviation for disturbance detection is a good, working approach when only using the bit error detection functionality. By using the eye width monitor, more advanced statistic analysis could be introduced. Unfortunately i lacked the time to further research this myself.

The implementation of a run module and a shutdown module worked well and is a way to limiting the amount of processing power needed in a ECU. The ECUs in modern cars are already occupied with their current tasks leaving no room to add processing heavy link diagnostics. Prolonging the heavy analysis to after shutdown of the car, when no other process is running is a way to include more features covered by a ECU without the need of upgrading the hardware, meaning no increase in production cost.

Working on this thesis have made me realize that there is a need for diagnostic applications like the SIVIC application in future cars. With the increased sensitivity due to higher data transfer speeds and the more trust we put in these system to drive us around, it is very important to know that every part of the system is working properly. It is not easy to simulate real world disturbances to test systems during development. Disturbances can vary dependent on the environment and a car is exposed to different EMI levels if it is driving along side a power line, inside a city or out on the country side. It is hard to know exactly what disturbances exists in these environment but there exist standards that specify how the more crucial disturbances should be generated and some of them are specified in the standard RI 130. RISE allowed me to use one of their labs and equipment to induce the disturbances covered by RI 130. By using this equipment, I was able to induce real transient disturbances to the test rigs and test how they are affected and understand if the SIVIC application detects any disturbances. From the tests where i used a cable with a broken shield, the data link didn't work at all. With data links of this type, they may be so sensitive that a small break in the shield renders the entire data link useless. It is then not useful to analyze if the cable is slowly breaking more and more by using a diagnostic software. Software like the SIVIC application can instead be used for detection of high disturbances emitted by other adjacent systems in a car. If some system starts emitting to much disturbances all of a sudden. Then maybe some other systems will detect it also.

During the tests at RISE some interesting test results evolved. The parking assist camera test rig was hardly affected by the disturbances compared to the evaluation board test rig which were certainly affected. Both of them used the same cable and were exposed to the same test sequence. By interviewing an expert of SerDes based data links, it appears as if the test rigs don't have the same ability to filter out disturbances in the cable shield which is why the different sensitivities occur. Neither of the test rigs are designed to be EMI resistant making them extra susceptible to disturbances. The data links used in production wouldn't be affected by these disturbances since they are designed to be more resistant.

The built in error generation feature is a good tool while working with the GMSL data links. It can be used to get to know how the link reacts to disturbances and it could be used to evaluate how well different implementations work, all without having access to lab environment and disturbance generating equipment. This feature was a widely used throughout the development phase as I had a limited amount of lab time. I only had access to a lab for one day to inflict real disturbances on the link. With more access to a lab, more analysis of how the data link reacts to disturbances could have been researched.

It was hard to determine the thresholds for when the data link is not credible any more. To get a kind of indication of how many bit errors the link could handle without affecting the video data content, the parking assists camera test rig were exposed to disturbances on different levels. While analyzing the video stream I noticed some small fractures in the picture at around 120 bit errors per second. However the systems using the video stream may be able to handle a higher amount of fractures in the picture and then can have a higher threshold.

All requirements but one was reached. Unfortunately i didn't succeed in changing the equalization and preemphasis levels to counteract the disturbances induced on the data link. I believe that the failure to reach this goal is due to the narrow time limits of the thesis. The SIVIC application, however, is able to perform it's main tasks and can be seen as a working data link diagnostic application.

The contents of this thesis could be used as a foundation for future work. Future researchers could look at the possibility to add a software like the SIVIC application to existing ECUs in cars. The SIVIC application could also be modified to work with other data links such as APIX of FPD-Link. The disturbance detection and trend analysis can also be further researched to produce more reliable and advanced methods by using the eye width monitor. One other topic which can be expanded is the data storage solution. With the increased amount of data it is important to have a good approach to structure and store it. A possible solution could be to upload it to a cloud service.

8
Conclusion

The evaluated SIVIC concept proves a way to use SerDes signals to detect and predict data link failure. The SIVIC application is able to fullfill the requirements set up in the concept development except for one, the temporary link remedy. However this is not a crucial functionality of the software, making the SIVIC application still able to accomplish its purpose of analyzing the link quality. The chosen statistical methods are able to detect the different types of disturbances affecting the system and the trend analysis is able to detect changes in the disturbance levels over time. The fault detection part of the SIVIC application is able to detect when a data link failure occurs.

The theoretical contribution of this thesis addresses 11 requirements that solves 11 aspects and provides a solution valid for any SerDes link. The practical contribution is the implementation and evaluation of SIVIC, and its adaptability for a short term industrial implementation.

9
Bibliography

[1] V. Vaishnavi and B. Kuechler, "Design research in information systems." http://desrist.org/design-research-in-information-systems, 2004. [Online, Accessed 3-April-2017].

[2] M. Integrated, "28-bit gmsl deserializer for coax or stp cable." https://www.maximintegrated.com/en/products/interface/high-speed-signaling/MAX9272A.html. [Online, Accessed 17-April-2017].

[3] C. Perform, "A comparison of cml and lvdsfor high-speed serial links." http://www.cypress.com/file/74011/download, 2002. [Online, Accessed 17-April-2017].

[4] M. Peffers, "Get connected: Serdes demystified." https://e2e.ti.com/blogs_/b/analogwire/archive/2014/05/08/get-connected-serdes-demystified, 2014. [Online, Accessed 4-April-2017].

[5] K. B. Kevin Sidwar, "A web developer's guide to communication protocols (spi, i2c, uart, gpio)." https://tessel.io/blog/108840925797/a-web-developers-guide-to-communication-protocols, 2015. [Online, Accessed 17-April-2017].

[6] IBM-Redbooks, "Eye diagrams," in *Understanding Optical Communications*, IBM Coorporation, 1998.

[7] Tech-Faq, "The osi model – what it is; why it matters; why it doesn't matter.." http://www.tech-faq.com/osi-model.html, 2016. [Online, Accessed 17-April-2017].

[8] M. Integrated, "Max96705 16-bit gmsl serializer with high-immunity/bandwidth mode and coax/stp cable drive." https://datasheets.maximintegrated.com/en/ds/MAX96705.pdf. [Online, Accessed 17-April-2017].

[9] "FMC1278electromagnetic compatability specification for electrical/electronic components and subsystems," standard, Ford, Oct. 2016.

[10] "ISO 26262-6:2011 road vehicles- functional safety- part 6: Product development at the software level," standard, Swedish Stadard Institute, Stockholm, Sweden, Nov. 2011.

[11] R. Gutzmer, "Video interface technology," in *Handbook of Camera Monitor Systems* (A. Terzis, ed.), pp. 235–275, Springer International Publishing Switzerland, 2016.

[12] M. Integrated, "Gigabit multimedia serial link (gmsl) serdes ics." https://www.maximintegrated.com/en/products/interface/high-speed-signaling/gmsl.html, 2017. [Online, Accessed 7-March-2017].

[13] "Maxim integrated." https://www.maximintegrated.com/en.html.

[14] Tech-FAQ, "What is the function of serdes?." http://www.tech-faq.com/serdes.html, 2016. [Online, Accessed 4-April-2017].

[15] C. University, "Lab 2 - 8b/10b encoder." http://www.cs.cornell.edu/courses/cs3410/2013sp/lab/lab2.html, 2013. [Online, Accessed 4-April-2017].

[16] "Inova semiconductor." https://www.inova-semiconductors.de/en/.

[17] "Texas instruments." http://www.ti.com/.

[18] F. Durda, "Serial and uart tutorial." https://www.freebsd.org/doc/en/articles/serial-uart/, 2014. [Online, Accessed 17-April-2017].

[19] C. Yilmazer, "An introduction to preemphasis and equalization in maxim gmsl serdes devices, howpublished =."

[20] V. Telang, "Equalization for high-speed serdes, howpublished =."

[21] A. Corporation, "Understanding eye pattern measurements." https://dl.cdn-anritsu.com/en-us/test-measurement/files/Application-Notes/Application-Note/11410-00533.pdf. [Online, Accessed 27-May-2017].

[22] M. Integrated, "Max9288 3.12gbps gmsl deserialziers for coax or stp input and mipi csi2 output." https://datasheets.maximintegrated.com/en/ds/MAX9288-MAX9290.pdf. [Online, Accessed 24-May-2017].

[23] M. Integrated, "Max9291/max9293 3.12gbps gmsl serializers for coax or stp output and hdmi input." https://datasheets.maximintegrated.com/en/ds/MAX9291-MAX9293.pdf. [Online, Accessed 25-May-2017].

[24] M. Integrated, "Max96706 14-bit gmsl deserializer with coax or stp cable input." https://datasheets.maximintegrated.com/en/ds/MAX96706.pdf. [Online, Accessed 25-May-2017].

[25] "Rise." https://www.ri.se/.

[26] emtest, "Rcb 200n1 transient immunity test generator meeting ford fmc1278 and emc-cs-2009.1." http://www.emtest.com/products/product/135120100000011403.pdf. [Online, Accessed 23-May-2017].

A
Appendix 1

Transient pulses and timing sequences possible to generate with RCB200N1 [26].

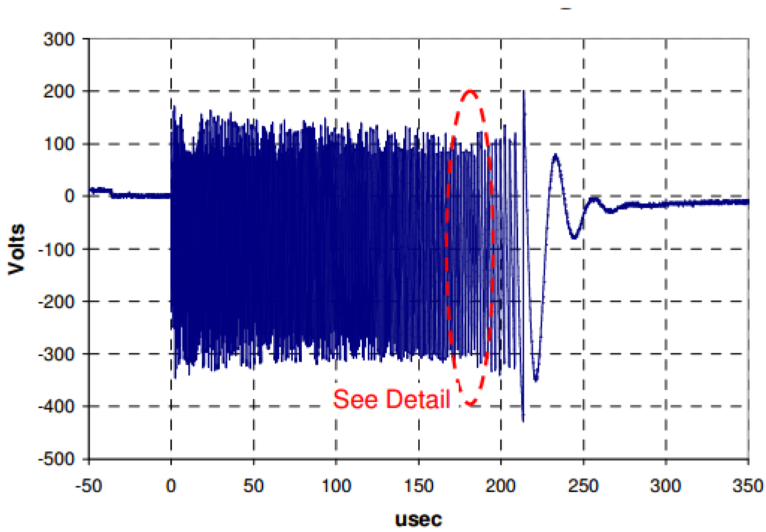

Figure A.1: Pulse A2-1 Contact Break and Bounce from [9].

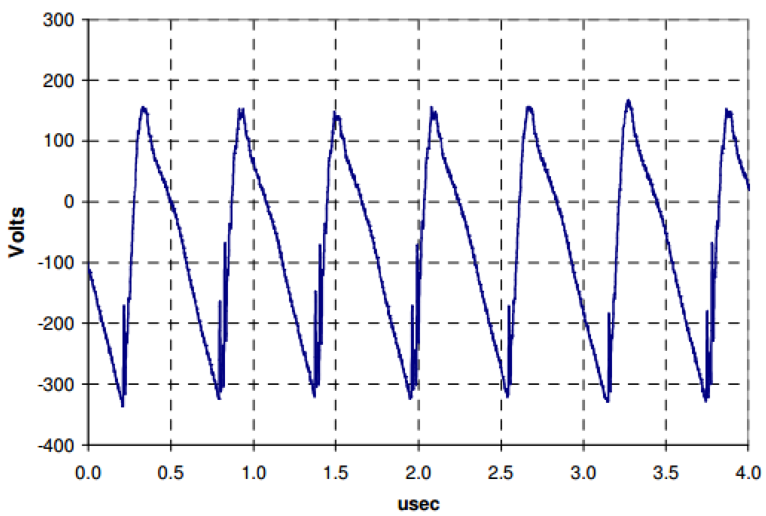

Figure A.2: Pulse A2-1 in detail. Contact Break and Bounce from [9].

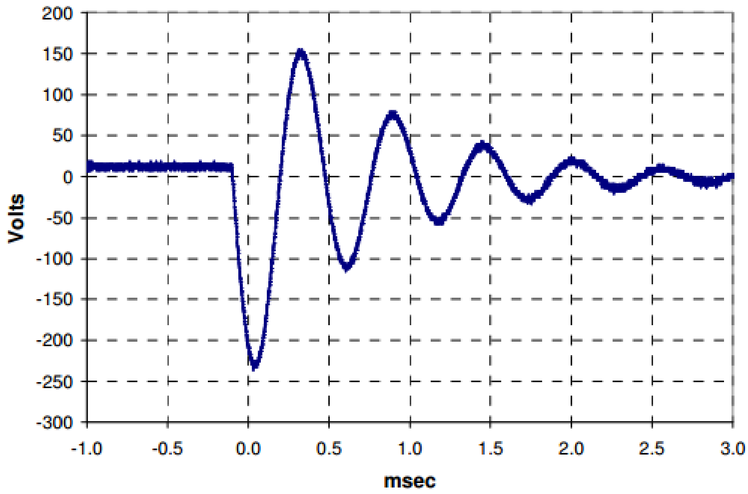

Figure A.3: Pulse A2-2 Contact Break from [9].

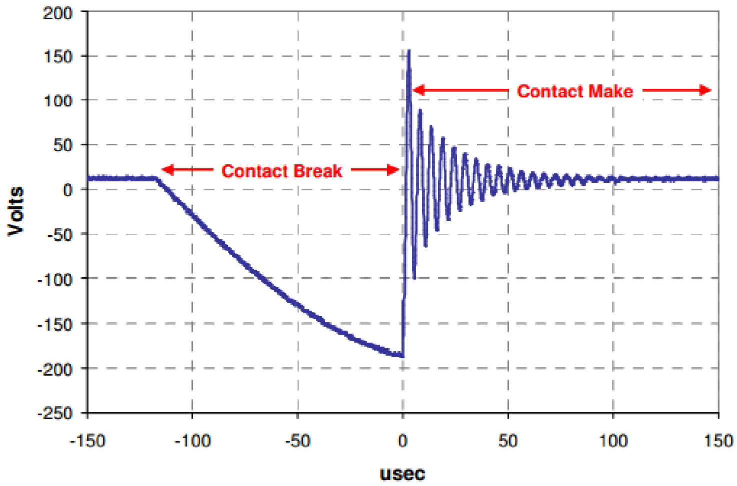

Figure A.4: Pulse A2-2 Contact Bounce (Voltage) from [9].

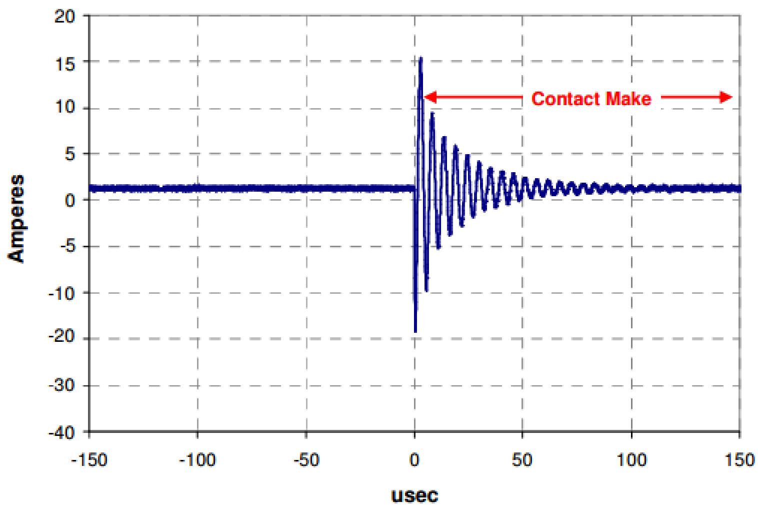

Figure A.5: Pulse A2-2 Contact Bounce (Current) from [9].

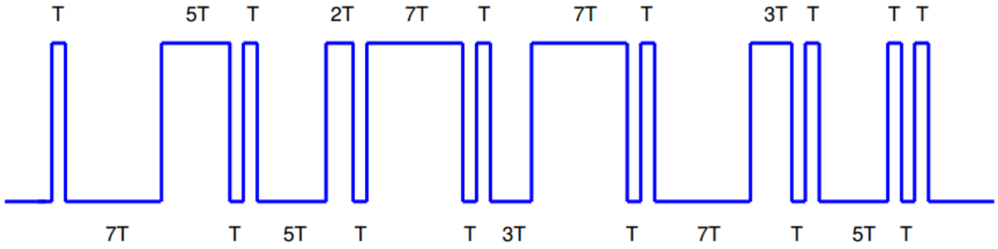

Figure A.6: Pulse timing sequence from [9]. T = 50 ms. (M2)

Figure A.7: Resulting transient sequence from [9]. T = 50 ms.

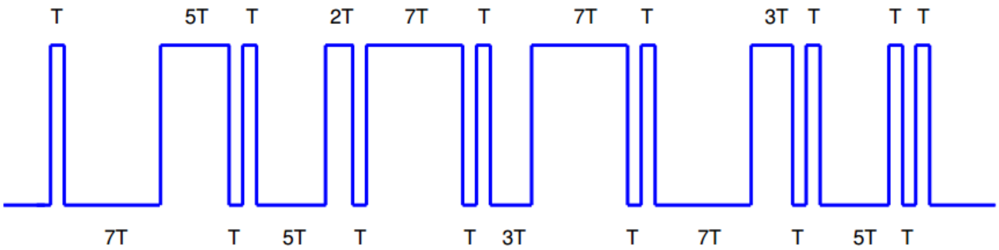

Figure A.8: Pulse Burst timing sequence from [9]. T = 50 ms. (M3)

Figure A.9: Resulting transient Burst from [9]. T = 50 ms.