Bit Error Rate Testing Serial Communication Equipment using Pseudo-Random Bit Sequences

Marius Strobl, Thomas Waas
University of Applied Sciences Regensburg
Regensburg, Germany

Marcel Moolenaar
Juniper Networks, Inc.
Sunnyvale, CA, USA

Angelika Schingale, Norbert Balbierer
Continental Automotive GmbH
Regensburg, Germany

***Abstract*—Although serial communication as in RS-232 is venerable by today's standards, it is still found in new designs. Moreover, unlike as in the old days when used to communicate over lossy phone lines, for instance in an automotive environment it is crucial to ensure its reliability and safety. In this paper, an approach for the necessary Bit Error Rate (BER) testing using Pseudo-Random Bit Sequences (PRBS) and borrowing from the IEEE 802.3 Ethernet standard is presented. The result of this research is a Bit Error Rate Tester (BERT), suitable to be included within the embedded devices composing the communication equipment.**

***Index Terms*—RS-232 serial communication, bit error rate testing, PRBS31, Yamar SIG60, automotive electronics**

I. Introduction

Back in the days when RS-232 [1] was the state of the art, it was mainly used to connect Teletypewriters (TTYs) as terminals to mainframes and to interconnect both of these using modems and phone lines. The reliability and safety of this serial communication was not much of a concern and the parity bit typically not used. Even the High-level Data Link Control (HDLC) protocol, as standardized by ISO/IEC 13239 [2] and commonly used on top of serial communication, advises to discard frames received with a checksum error, mostly silently [2, p. 58]. RS-232 is still used as of today for providing serial consoles to headless servers and to network equipment, but bit errors again are not a problem there. However, when used as part of an automotive network, reliability and safety of serial communication equipment suddenly are important.

The devices leading to the research presented in this paper are the SIG60 [3] Powerline Communication (PLC) transceivers from Yamar Electronics Ltd. These are automotive components providing digital communication at data rates of up to 115.2 kbps over vehicular battery powerline. They are designed to be interfaced with the Universal Asynchronous Receiver Transmitter (UART) of a host via RS-232. As part of a research project investigating the possibilities to replace the "low speed" Controller Area Network (CAN) and Local Interconnect Network (LIN) in an "All IP/Ethernet Car" [4, p. 17], Ethernet-PLC-bridges further described in [5] have been built upon the SIG60. These bridges communicate over the powerline, presenting themselves as a distributed hub towards the nodes connected to them via an IEEE 802.3 [6] compliant Ethernet interface. As the Ethernet-PLC-bridges are intended for use in an automotive network, it was crucial to be able to test the Bit Error Rate (BER) of the PLC modems both in their development phase and later on to determine the influence of powerline attenuation on the quality of the communication.

The BER is the unitless ratio of the number of bit errors divided by the total number of transferred bits during a given time interval [7].

According to information provided by the vendor, the BER expected for SIG60 equipped with ceramic filters at a data rate of 57.6 kbps and a line attenuation of up to 3 dB would be in the magnitude of 10^{-7} or less. Above that, every additional dB would increase the BER by one order of magnitude until a loss of communication with an attenuation beyond 6 dB.

II. Bit error rate testing

Given that Ethernet-PLC-bridges were to be built, it was obvious to use the method employed for the BER testing of Ethernet also for the PLC modems.

A. Test pattern generator

IEEE 802.3 suggests [6, section four, p. 269 ff.] the test pattern generator shown in Figure 2 for link verification. The output of this generator is the Pseudo-Random Bit Sequences (PRBS) of order 31 (PRBS31), which is the *inverted* bit stream produced by (1).

$$G(x) = 1 + x^{28} + x^{31} \tag{1}$$

The advantage of using a PRBS pattern for BER testing is that it is a deterministic signal with properties similar to those of a random signal for the link [8], i. e. of white noise. Moreover, the sequence being pseudo random, both the transmitter and the receiver can produce it locally using their own PRBS31 generator. In order to synchronize its local pattern generator with the transmitter, the receiver uses the first part of the bit stream produced by the transmitter for initialization. Given that this first part was received correctly, both PRBS31 generators now are in sync and the receiver may compare the transmitter's further sequence to its locally generated stream for determining bit errors caused by the communications equipment or the link. The basic concept of such a PRBS31-based Bit Error Rate Tester (BERT) is shown in Figure 1.

In general, the specification of the IEEE 802.3 PRBS31 pattern generator shown in Figure 2 lends itself to being implemented using a Linear Feedback Shift Register (LFSR). As seen in this figure, an LFSR is a shift register which feeds back the output of one state into the input of the next one.

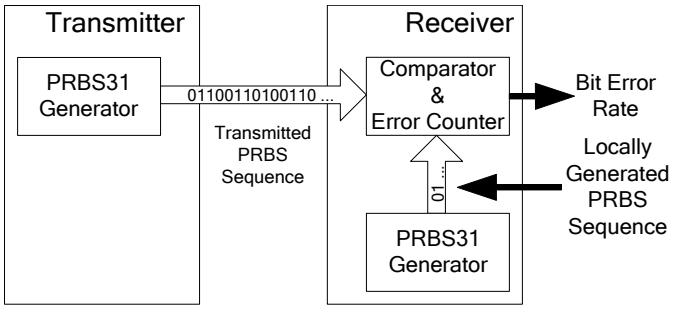

Figure 1. PRBS31-based Bit Error Rate Tester (BERT) (based on [8])

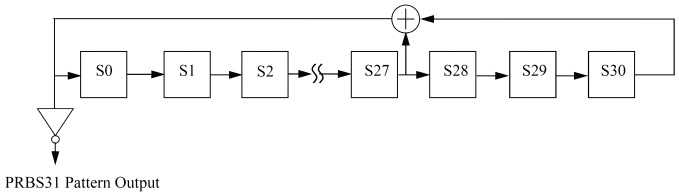

Figure 2. IEEE 802.3 PRBS31 pattern generator [6, section four, p. 271]

Therefore, the output is a linear function of the previous input. Due to the fact that an LFSR has a finite number of states determined by its length, it implements a repeating cycle of deterministic states, unless a state contains all zeros. In the latter case, the LFSR does not change the state. This likely also is the reason for IEEE 802.3 stating to seed the PRBS31 pattern generator with an initialization vector different from zero. When built upon the maximum-length polynomial for the shift register of length n, an LFSR cycles through a maximum-length sequence (MLS), i. e. produces all possible $2^n - 1$ states. The bit positions influencing the next state are called taps. The PRBS31 generator polynomial in (1) is the maximum-length polynomial for 31 bits using the taps 28 and 31. A table containing the characteristic polynomials for maximal LFSRs of the length 3 to 168 bits can be found in [9, p. 5].

As shown by A. Rasmussen *et al.* in [8], when implemented in hardware, PRBS31-based link testing is even suitable for 100 Gbps Ethernet equipment. For the implementation of the PRBS31 pattern generator in the C language shown below, a Galois LFSR has been chosen though. This corresponding function produces the bit sequence of one state on every call.

```
#define PRBS_IV 0x1

/*
 * PRBS31 generator according to IEEE 802.3-2008 49.2.8
 * polynomial: G(x) = 1 + x^28 + x^31, output inverted
 */
static uint32_t
prbs(void)
{
        static uint32_t lfsr = PRBS_IV;

        lfsr = (lfsr >> 1) ^ (-(lfsr & (uint32_t)1) &
            (((uint32_t)1 << 30) | ((uint32_t)1 << 27)));
        return (~lfsr);
}
```

Not counting the inversion, this implementation uses one `OR` (unless there is only one tap) and one `SHIFT` per tap plus 5 operations per state. That one `SHIFT` per tap has been added for the sake of readability and can be optimized out by the compiler or the result hardcoded in the first place, making up a net total of 7 operations per call. This LFSR implementation works by creating a mask using the two's complement negation on the least significant bit of the feedback. The result of `AND`ing this mask with the tap values is a toggle mask which determines whether to apply the tap mask based on the feedback bit. This toggle mask then is `XOR`ed with the `SHIFT`ed value of the input, creating the new output.

The other common implementation not shown here is a Fibonacci LFSR. The latter use one `SHIFT` and one `XOR` per tap plus 4 operations per state. So with 7 versus 8 operations per state, using a Galois LFSR already yields a net win with the 2 taps sought and scales even better with additional taps.

Given that the receiver also uses the same PRBS31 pattern generator locally, detecting bit errors in the stream typically transferred byte-wise via the serial communication equipment is as simple as directly comparing received bytes with the locally produced ones. However, when writing a BERT application (see section III), the byte order of the processor running the application should be taken into account when converting the 32-bit values of the `prbs()` function into bytes for transmission and vice versa. Doing so allows transmitter and receiver to be of different endiannesses.

B. Bit error counting

Whereas a mask of the bit errors in the stream can be created by `AND`ing the received bytes after coalescing them with the locally generated PRBS31 pattern, counting the number of bits set in this mask in order to calculate the BER is a bit tricky. For this, the parallel bit counting implementation shown below has been chosen. It is part of S. E. Anderson's excellent Bit Twiddling Hacks collection for the C language [10].

```
/*
 * Count bits set, in parallel
 */
static u_int
bitcount(uint32_t val)
{

        val = val - ((val >> 1) & 0x55555555);
        val = (val & 0x33333333) + ((val >> 2) &
            0x33333333);
        return ((((val + (val >> 4)) & 0xf0f0f0f) *
            0x1010101) >> 24);
}
```

This algorithm is best explained by the author himself:

> *"The best bit counting method takes only 12 operations, which is the same as the lookup-table method, but avoids the memory and potential cache misses of a table. It is a hybrid between the purely parallel method above and the earlier methods using multiplies (in the section on counting bits with 64-bit instructions), though it doesn't use 64-bit instructions. The counts of bits set in the bytes is done in parallel, and the sum total of the bits set in the bytes is computed by multiplying by 0x1010101 and shifting right 24 bits."* — S. E. Anderson [10]

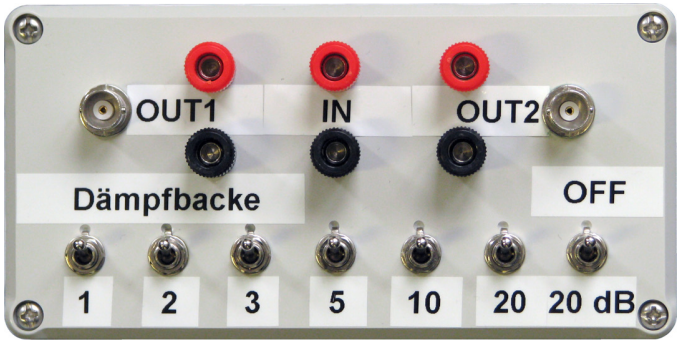

Figure 3. "Dämpfbacke" line attenuation simulator

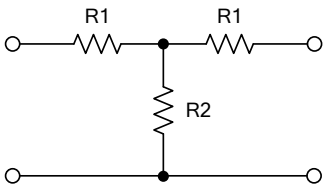

Figure 4. Asymmetric T pad attenuator schematic circuit

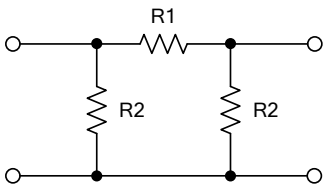

Figure 5. Asymmetric Π pad attenuator schematic circuit

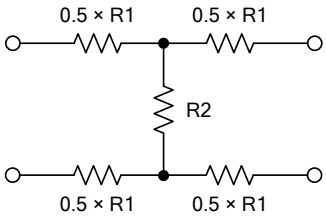

Figure 6. Symmetric T pad attenuator schematic circuit

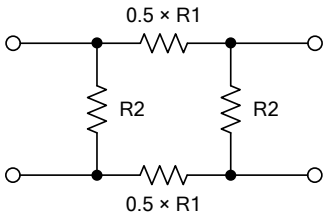

Figure 7. Symmetric Π pad attenuator schematic circuit

Another reason for not using a method based on a lookup-table for a fast implementation on embedded systems is that there such tables typically are kept in flash memory instead of copying them to the precious RAM. However, for the Atmel® AVR® microcontrollers for instance, the access times of flash memory may be longer than those of RAM [11, p. 4 f.].

III. Bit error rate tester application

Based on the methods described above, a BERT application for serial communication via UARTs and optionally SIG60 has been implemented, which is available at [12]. Originally, this BERT has been written for the Ethernut 1 [13] microcontroller reference design boards from egnite GmbH also employed in the Ethernet-PLC-bridges [5]. Hence, the objective of the implementation was suitability for embedded systems regarding execution constraints (see also [14]). However, for flexibility and ease of use, the version presented in this paper builds on the IEEE 1003.1 [15] Portable Operating System Interface (POSIX®) instead. As part of this research, it has been developed and used on FreeBSD [16, p. 113] as well as additionally run on GNU/Linux. This BERT should also work with any other POSIX® compatible operating system though. After compilation, this application may be called using the following command-line syntax:

```
/path/to/bert -[c|f|i|t] [-s speed] <device>
```

The command-line options are as follows:

- `-c` Receiver (data communication equipment)
- `-f` Select filter F1 (when use with `-i`).
- `-i` Initialize a Yamar SIG60.
- `-s` Data rate in bps
- `-t` Transmitter (data circuit-terminating equipment)

For `device`, the name of the callout device node corresponding to the serial interface to be used, for instance `/dev/cuau0`, should be passed. The data rate may be specified in bps supplying a value of 9600, 19200, 38400, 57600 or 115200 for the `speed` parameter. If the data rate option is omitted on the command-line, the BERT uses a default value of 9.6 kbps. However, for SIG60, the availability of individual data rates depends on the filters installed.

This BERT application expects the Data Terminal Ready (DTR) line of the serial interface used to be connected to the Host Data/Command (HDC) pin of the SIG60 for optionally initializing the latter before running the actual BERT. Thus, this BERT also does not interrogate any of the modem status lines in order to determine whether the other side is up.

When the application running in receiver mode is synchronized with the transmitter and receives any keypress, the time since the synchronization, the current number of all octets received, the number of bit errors and the BER are displayed. The BERT can be aborted by sending it the `SIGINT` signal.

IV. Line attenuation simulator

In order to be able to verify the claim of Yamar regarding the BER characteristics of SIG60 and to test the robustness of their PLC in general, the line attenuation simulator shown in Figure 3 has been built. The latter resembles a step switch consisting of a series of shiftable asymmetric T pad attenuators with 1, 2, 3, 5, 10 and twice 20 dB connected at their terminals. These sum up to the desired attenuation between the `OUT1`, directly connected to the `IN`, and the `OUT2` port of the simulator.

Generally, the asymmetric and symmetric T and Π pads shown in figures 4 through 7 are two-port network circuits commonly employed in telecommunications engineering for building attenuators. Their topology – asymmetric or symmetric – has to be chosen based on the intended application [17, p. 113]. While Π pad attenuators are better suited for purposes where the series inductance of the entire setup predominates, the T pad variant is advantageous if – as in the intended use case – the capacity towards ground outweighs [18, p. 41]. As shown in Figure 9, another benefit of asymmetric T pads for building a line simulator is that these can be made shiftable with a simple double pole, double throw (DPDT) switch.

Based on the impedance Z in [Ω] of the cabling and the whole system that has to be matched, the values R_1 and R_2 in [Ω] of the resistors to be used in an asymmetric T pad circuit of the desired attenuation a in [dB] can be calculated as shown

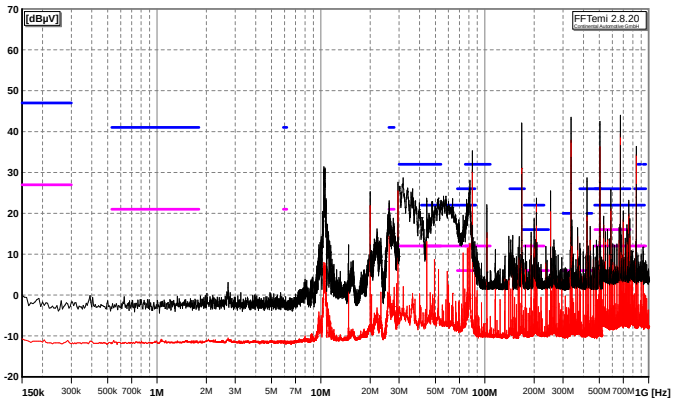

Figure 8. Peak (black) and average (red) stripline EMI spectra of two Ethernet-PLC-bridges using SIG60-based PLC modems communicating with a data rate of 115.2 kbps at a channel frequency of 10.5 MHz, IEC CISPR 25:2008 [20, p. 71 f.] stripline class 5 peak (blue) and average (purple) limits

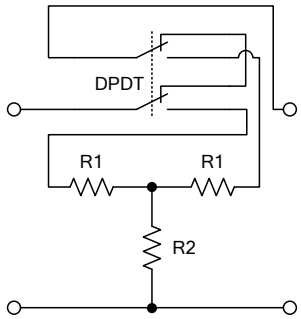

Figure 9. Schematic circuit of a switchable asymmetric T pad attenuator

in (2) through (4) by combining the formulae provided in [18, p. 157] and [18, p. 159].

$$\alpha = 10^{\frac{a}{-20}} \tag{2}$$

$$R_1 = \frac{1-\alpha}{1+\alpha} \times Z \tag{3}$$

$$R_2 = \frac{\frac{1+\alpha}{1-\alpha} \times Z - R_1}{2} \tag{4}$$

The asymmetric T pad attenuators of the line simulator have been built using the closest matching resistors existent in the available E series, based on their calculated values and targeting an impedance of 100 Ω as listed in table I. Metal foil resistors, which yield less current noise than carbon film ones [19], with a tolerance of 1 % have been chosen for this purpose. An impedance of 100 Ω thereby matches that of the ribbon conductor used for connecting the SIG60-based PLC modems in the laboratory testing environment.

V. Equipment under test

The SIG60-based PLC modems tested are prototypes built on the schematics example [3, p. 6]. However, instead of the recommended ceramic filters, these modems use discrete ones. Compared to the ceramic filters, the latter provide a broader

Table I
Resistor values calculated and used for the asymmetric T pad attenuators matching an impedance of 100 Ω as a function of the attenuation

Attenuation [dB]	R1 [Ω]		R2 [Ω]	
	calculated	used	calculated	used
1	5.75	5.62	866.67	866.00
2	11.46	11.50	430.48	430.00
3	17.10	17.40	283.85	280.00
5	28.01	30.00	164.01	160.00
10	51.95	52.30	70.27	68.00
20	81.82	82.00	20.20	20.00

Table II
Bit error rate magnitudes of reference tests and SIG60-based PLC modems as a function of data rate and channel frequency

Attenuation [dB]	BER at 57.6 kbps		BER at 115.2 kbps	
	10.5 MHz	13.0 MHz	10.5 MHz	13.0 MHz
direct	no bit errors			
0 ... 1	no bit errors	no bit errors	10^{-8}	10^{-7}
1			10^{-9}	10^{-8}
2			no bit errors	
3			10^{-8}	
4				10^{-7}
5				
6		byte loss	10^{-7}	byte loss
7				
8			byte loss	
9				
10	byte loss			

bandwidth around the channel frequencies of 10.5 MHz and 13.0 MHz required for data rates of 57.6 kbps and 115.2 kbps.

Although these modems are prototypes, as illustrated in Figure 8, they are already close to adhering to the IEC CISPR 25:2008 [20] constraints for automotive Electromagnetic Interference (EMI) up to class 5 of 5. Note that although said standard defines limits sectionally, these are generally interpreted as a continuous limit laid through the provided intervals for filling the gaps and using the lower limit in case of overlaps as done in e. g. [21]. This approach has been chosen because of the fact that differences in harness lengths found in reality may shift apexes in the EMI spectra from laying within such gaps into the defined intervals of these limits.

Actually, the main problem still left to be solved with these SIG60-based PLC modems solely are the resonances encountered in the stripline EMI spectra at approximately 84 MHz, 176 MHz and 252 MHz et cetera. These correspond to the 1.7 m length of the harness used in the Electromagnetic Compatibility (EMC) measurement setup and its harmonics as well as when the harness is acting as a Hertzian dipole.

VI. Results

The results of testing the SIG60-based PLC modems described in section V at data rates of 57.6 kbps and 115.2 kbps using the BERT introduced in section III and that of reference tests between two computers directly connected via a null-modem cable are listed in Table II. Herein, the attenuation of

0 … 1 dB denotes that no artificial attenuation besides the one caused by the cabling has been applied.

These tests have been conducted in a laboratory environment using the line attenuation simulator presented in section IV and a studied time interval of 1 hour per run. An upper limit of 10 dB for these measurements has been chosen because – although being of the low-dropout type – the linear voltage regulator used on the modems started to fail to supply the circuit with 3.3 V from the nominal 12 V level before the simulator at an attenuation around 11 dB.

Across multiple runs of the same test, the magnitudes of these BERs have been reproducible.

VII. Conclusion

In this paper, an approach for determining the BER of serial communication equipment has been presented. Due to the optimizations taken, it is suitable to be integrated within the software of embedded systems and that of the equipment itself. The reference implementation provided for POSIX® compatible operating system allows to test the BER via a serial interface and optionally that of Yamar SIG60 PLC transceivers.

As for the prototypes of SIG60-based PLC modem tested, the BERs measured are better than what Yamar had predicted. The enhanced robustness against line attenuation can be attributed to the increased bandwidth of the discrete filters employed compared to the recommended ceramic ones.

However, at least when using a data rate of 115.2 kbps, the BERs observed are reasonably consistent with the graduation envisioned by Yamar, with an optimum apparently reached at an attenuation of 2 dB. The latter suggest that actually the receiver part of the SIG60 is driven into saturation with insufficient line attenuation.

The worst case BER of 10^{-7} encountered with these modems also matches what is expected for CAN [22]. No references regarding the anticipated BER of the LIN bus could be found. A BER of 10^{-7} is also the worst case limit when using twisted pair *(sic)* cabling as communication medium regardless of the protocol layers involved [23, p. 4] though.

Acknowledgment

The research leading to the results presented in this paper is supported by “Regionale Wettbewerbsfähigkeit und Beschäftigung”, Bayern, 2007-2013 (EFRE) as part of the SECBIT project (http://www.secbit.de).

References

[1] *Interface between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange*, EIA Std. RS-232-C, 1969.

[2] *Information technology – Telecommunications and information exchange between systems – High-level data link control (HDLC) procedures*, ISO/IEC Std. 13 239, 2002.

[3] “SIG60 – UART over Powerline, for AC/DC-BUS Network,” Datasheet, Yamar Electronics Ltd., Tel Aviv, Israel, 2012. [Online]. Available: http://www.yamar.com/datasheet/DS-SIG60.pdf

[4] H. Zinner, “Ethernet AVB als Backbonenetzwerk auf Basis eines QoS Gateways – Application and Realization of Gateways between conventional Automotive and IP Ethernet based Network,” presented at the SEIS Statusseminar, Munich, Germany, Sep. 2011. [Online]. Available: http://www.strategiekreis-elektromobilitaet.de/public/projekte/seis/das-sichere-ip-basierte-fahrzeugbordnetz/pdfs/TP2_Vortrag2.pdf

[5] M. Strobl *et al.*, “Using Ethernet over Powerline Communication in Automotive Networks,” in *Proc. IEEE Tenth Workshop on Intelligent Solutions in Embedded Systems*, Klagenfurt, Austria, Jul. 2012, pp. 39–44. [Online]. Available: http://wises2012.nes.aau.at/fileadmin/papers/submission7.pdf

[6] *Information technology – Specific requirements – Part 3: Carrier sense multiple access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications*, IEEE Std. 802.3, 2008. [Online]. Available: http://standards.ieee.org/about/get/802/802.3.html

[7] G. Breed, “Bit Error Rate: Fundamental Concepts and Measurement Issues,” *High Frequency Electronics*, vol. 2, no. 1, pp. 46–47, Jan. 2003. [Online]. Available: http://www.highfrequencyelectronics.com/Archives/Jan03/HFE0103_Tutorial.pdf

[8] A. Rasmussen *et al.*, “Framed Bit Error Rate Testing for 100G Ethernet Equipment,” in *Proc. IEEE International Conference on High Performance Switching and Routing (HPSR)*, Dallas, TX, USA, Jun. 2010, pp. 165–168.

[9] “Efficient Shift Registers, LFSR, Counters, and Long Pseudo-Random Sequence Generators,” Application Note, Xilinx, Inc., San Jose, CA, USA, Jul. 1996. [Online]. Available: http://www.xilinx.com/support/documentation/application_notes/xapp052.pdf

[10] S. E. Anderson, “Bit Twiddling Hacks, Counting bits set, in parallel,” 2005. [Online]. Available: http://graphics.stanford.edu/~seander/bithacks.html#CountBitsSetParallel

[11] “Nut/OS Memory Considerations,” egnite Software GmbH, Castrop-Rauxel, Germany, 2007. [Online]. Available: http://www.ethernut.de/pdf/enmem21e.pdf

[12] M. Strobl. (2012, Mar.) Bit error rate tester (BERT) for serial communication via UARTs and optionally Yamar SIG60. [Online]. Available: http://www.freebsd.org/~marius/research/bert.tar.bz2

[13] Ethernut version 1.3 hardware. Website. egnite Software GmbH. Castrop-Rauxel, Germany. [Online]. Available: http://www.ethernut.de/en/hardware/enut1/index.html

[14] T. Henzinger and J. Sifakis, “The embedded systems design challenge,” in *Proc. 14th International Symposium on Formal Methods (FM), Lecture Notes in Computer Science*. Hamilton, Ontario, Canada: Springer-Verlag, Aug. 2006, pp. 1–15.

[15] *Information technology – Portable Operating System Interface (POSIX®) – Base Specifications, Issue 7*, IEEE Std. 1003.1, 2008.

[16] The FreeBSD Project. Website. The FreeBSD Foundation. Boulder, CO, USA. [Online]. Available: http://www.freebsd.org

[17] H. Cordes, *Praxiserprobte Messtechnik für Funkamateure, Grundlagen, Theorie, Praxis*. Baden-Baden, Germany: Verlag für Technik und Handwerk, 1994.

[18] H. Heuermann, *Hochfrequenztechnik, Komponenten für High-Speed- und Hochfrequenzschaltungen*, 2nd ed. Wiesbaden, Germany: Vieweg+Teubner, GWV Fachverlage GmbH, 2009.

[19] T. Williams and J. Thomas, “A Comparison of the Noise and Voltage Coefficients of Precision Metal Film and Carbon Film Resistors,” *IRE Transactions on Component Parts*, vol. 6, no. 2, pp. 58–62, Jun. 1959. [Online]. Available: http://dalmura.com.au/projects/Resistor%20noise.pdf

[20] *Vehicles, boats and internal combustion engines, Radio disturbance characteristics – Limits and methods of measurement for the protection of on-board receivers*, IEC Std. CISPR 25, 2008.

[21] F. Nouvel and P. Tanguy, “What is about Future High Speed Power Line Communication Systems for In-vehicles Networks?” in *Proc. IEEE 7th International Conference on Information, Communications and Signal Processing*, Macau, China, Dec. 2009, pp. 1–6. [Online]. Available: http://hal.archives-ouvertes.fr/docs/00/45/63/86/PDF/ICICS2009_version_soumise.pdf

[22] J. Ferreira, A. Oliveira, P. Fonseca, and J. Fonseca, “An Experiment to Assess Bit Error Rate in CAN,” in *Proc. 3rd International Workshop of Real-Time Networks (RTN2004)*, Catania, Italy, Jun. 2004, pp. 15–18. [Online]. Available: http://www.ieeta.pt/lse/rtn2004/preprints/15-prep.pdf

[23] A. Schön, “Und am Anfang war das Ethernet … die Unterlage des Internet,” presented at the Kommunikationsnetz Franken e. V. Kongress 2001, Nuremberg, Germany, Nov. 2001. [Online]. Available: http://www.franken.de/fileadmin/mediapool/kongress/2001/ethernet.pdf