

OFDM BASED DATA LINK FOR THE DLR RESEARCH AIRCRAFT ATRA

*Nicolas Schneckenburger, Christoph Klein, Michael Schnell,
German Aerospace Center (DLR), Oberpfaffenhofen, Germany*

Abstract

In this paper the new broadband data link radio (B-DLR), developed for the German Aerospace Center (DLR) research aircraft is presented. The link shares strong similarities with the L-band communication system L-DACS1, also partly developed at DLR. The paper presents details concerning both the design and implementation of B-DLR. This includes a description of the physical layer as well as all protocol functionalities implemented in B-DLR in a single upper layer. Additionally, details about the requirements on the hardware and software necessary in order to set up a bi-directional data transmission in real time are given.

Introduction

Currently, the Advanced Technology Research Aircraft (ATRA) is going into service as a new flight test platform of DLR based on an Airbus A320-232. One important requirement is to allow real time measurement data analysis on the ground. Thus, in 2008 the Institute of Communication and Navigation started the development of a new robust, high bandwidth data link, named B-DLR, used for the bi-directional exchange of information between the flying aircraft and the ground station.



Figure 1. ATRA landing at DLR in Braunschweig.

Recently, a large amount of research at DLR is being conducted on the topic of future aeronautical communication systems. In order to cope with the increasing demand of communication capacity in the aeronautical sector, the Future Communications Infrastructure (FCI) has been developed by FAA and Eurocontrol. DLR is involved in the development and standardization of one of the two candidates now considered for deployment, the L-band digital aeronautical communication system candidate 1 (L-DACS1) [1,2]. Currently the development is going on under the SESAR framework in project P15.2.4 “Future Mobile Data Link System Definition” [3].

Due to the experience with the future L-band aeronautical communication system, the new B-DLR system is designed to have strong similarities with L-DACS1. Since B-DLR is designed as a single user system, major simplifications compared to the multi-user L-DACS1 system can be made. This especially applies to the upper layer protocol, where the implementational complexity can be reduced significantly.

The actual implementation of a simplified, but otherwise complete aeronautical communication system serves two purposes for DLR: First a robust, high data rate, broadband link is available during flight time to transfer measurement data to the ground and vice versa. Secondly, and equally important, a proof of concept for the L-DACS1 system is given. Although different bandwidths and a different carrier frequency (S-band) are used, the physical layer parameters for both systems are almost identical. A demonstration of transferring data between the ATRA and the ground station using the B-DLR transmitter will lead to important findings, which could then be transferred for future tests of L-DACS1. This is especially true, since B-DLR can be seen as a simplified L-DACS1 system in the S-band.

The paper is organized as follows: First a short overview of the most important B-DLR parameters is given. This includes the modulation and channel coding parameters as well as a brief description of B-DLR’s upper layer transmission protocol. The second section deals with the implementation of the

transmitter and receiver describing the hardware and software necessary for setting up the entire system. In the last section an evaluation of the system's performance in terms of area of coverage as well as transmission rate is given. The paper closes with a conclusions and an outlook about future work.

Overview of B-DLR Parameters

In this section, a brief overview over the physical layer parameters of the communication system is given. The changes from L-DACS1 to B-DLR are highlighted in order to stress the slight differences between both systems. The section starts with presenting the general B-DLR transmission characteristics. This is followed by the processing chain of the physical layer. The section closes with a brief overview over the upper layer protocol employed in B-DLR.

Unlike L-DACS1, which transmits in the L-band, for B-DLR frequencies in the S-band were assigned to the DLR in the October, 2009. The German Federal Network Agency (Bundesnetzagentur) has allocated two frequency bands of 5 MHz and 7 MHz bandwidth in the S-band with a transmit power of 5 Watts for experimental radio transmissions. To have symmetric capacities for both the uplink and the downlink, $\Delta f_{\text{total}} = 5 \text{ MHz}$ are used in both directions. This marks a major difference between B-DLR and L-DACS1. While L-DACS1 has a comparably low bandwidth of 500 kHz, the spectral resources for ATRA's communication system are increased by a factor of ten.

As for L-DACS1 the modulation is orthogonal frequency-division multiplexing (OFDM). With an identical subcarrier spacing of $\Delta f_s \approx 9.8 \text{ kHz}$ compared to L-DACS1 [2], the size of the fast Fourier transform (FFT) is increased to a nominal length of $N_{\text{FFT}} = 512$ in order to fill the available spectrum. With the DC carrier and 60 and 61, respectively, guard carriers on each side left empty, in total 410 carriers are available for the transmission of payload data and control information. This leads to an effective bandwidth of $\Delta f_{\text{useful}} \approx 4.014 \text{ MHz}$. The allocation of the carriers is illustrated in Figure 2.

Unlike L-DACS1 where several transmission modes and signaling channels exist, in B-DLR only the one frame type shown in Figure 2 is specified.

After an AGC preamble, two synchronization OFDM symbols are inserted. These two symbols also carry control information. The synchronization symbols are followed by 11 OFDM symbols designated for payload data. This results in a total number of 4510 modulation symbols to be transmitted in every frame. Of these symbols only 4198 may be used to transmit payload data due to the insertion of pilots used for channel estimation. The modulation alphabet is currently limited to QPSK. However an extension to higher order alphabets is possible and planned for the future.

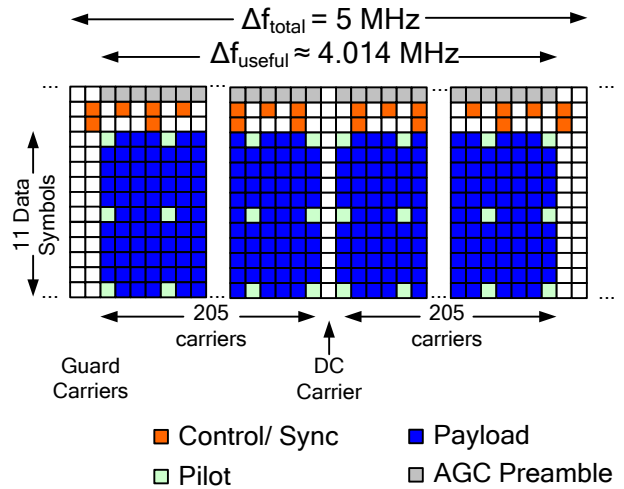


Figure 2. Frame structure of B-DLR.

In Figure 3 the composition of an OFDM symbol in the time domain is shown. The gradient at the bottom of the figure illustrates which parts of the signal are copied for the cyclic prefix (CP) and windowing function. Each OFDM symbol with the useful symbol duration of $t_{\text{useful}} = 102.4 \mu\text{s}$ (512 samples) is extended into a cyclic prefix of length $t_{\text{cp}} = 4.8$ (24 samples). Additionally a raised cosine windowing function is applied to each OFDM symbol in order to reduce out of band radiation. This adds another $t_{\text{win}} = 12.8 \mu\text{s}$ (64 samples) on each side of an OFDM symbol. However, due to the overlapping of the windowing function between the consecutive symbols the overall signal duration is $t_{\text{symbol}} = 120 \mu\text{s}$ (600 samples). Thus the overall CP and windowing overhead is about 15%.

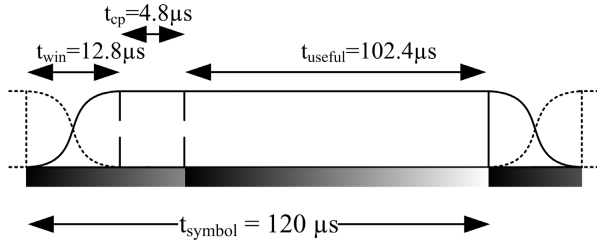


Figure 3. B-DLR OFDM symbol.

Physical Layer Processing Chain of B-DLR

In the following section the operations performed on the payload and control bits in order to obtain the final transmit signal are described. The entire processing chain of B-DLR is shown in Figure 4.

To both groups of data first a cyclic redundancy check (CRC) of length 32, for the payload data, and 16 bits, for the control data, is added. The main task of the scrambling block is to reduce the transmit signal's peak to average power ratio (PAPR) for long sequences of consecutive ones or zeros.

The first difference between the processing of the control and payload bits becomes obvious at the forward error correction: For the payload data the same coding scheme as for L-DACS1 is employed. First an outer Reed-Solomon RS(131, 119, 6) code ($r_{RS,p} \approx 0.9$) is applied. This results in 4 RS code words. The code words are then interleaved on symbol level basis. The result is encoded using a zero terminated inner convolutional code of rate $r_{CC,p} = 0.5$ and a memory length $m_{CC,p} = 6$. The interleaving on a symbol level between the two encoders assures a distribution of the convolutional decoder's errors on different RS code words. The overall coding rate of the scheme for the payload data is $r_p \approx 0.45$.

The control data is encoded using a lower code rate in order to allow correct decoding in significantly lower carrier to noise ratio (C/N) regions. This is because lost or incorrect control information decreases the overall system's performance drastically. The outer RS code is replaced by a zero terminated convolutional code of rate $r_{CC,c} = 0.25$ with a memory length of $m_{CC,c} = 5$. The output of the convolutional coder is then repeated 3 times, i.e. a repetition code of rate $r_{RP,c} \approx 0.33$ is employed. The resulting overall coding rate for the control data is $r_c \approx 0.08$. The coding

scheme for the control bits was designed in order to minimize the implementational complexity. Obviously a more sophisticated scheme, e.g. exchanging the inner repetition code with the outer convolutional code leads to a better performance of the resulting code. However for efficiently decoding a code like this, different decoders as the ones already available would need to be implemented.

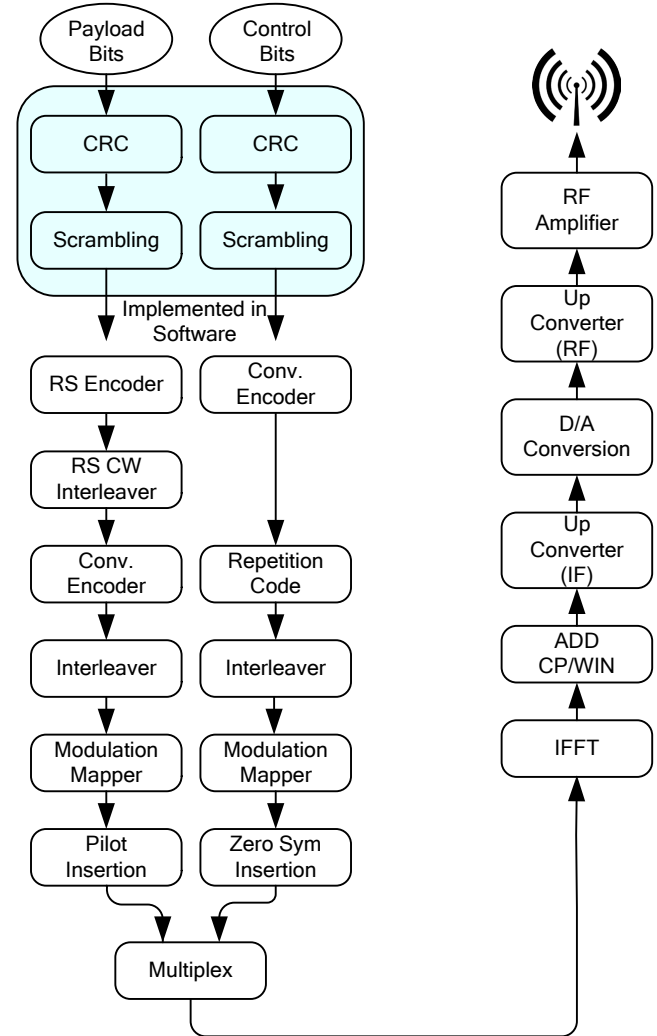


Figure 4. Processing chain of B-DLR.

The forward error protection is followed by a rectangular block interleaver. The bits are written into a matrix row wise, an inter column permutation is performed and the data is read out column wise. Depending on the modulation alphabet each stream is then split into groups of bits and mapped to complex symbols. In a first stage both the control and the

payload data are mapped on a QPSK alphabet. However, in the hardware design also 16QAM and 64QAM are possible. Into the stream of modulated payload symbols additionally pilots are inserted according to Figure 2.

The coded control data symbols are not interleaved into the stream of payload data, as often seen in other communication systems, e.g. LTE. In B-DLR the control bits precede the actual payload data. Using a special alignment of the control symbols in the frequency domain, the control information acts as a synchronization sequence for the entire frame as illustrated in Figure 2. This is possible due to B-DLR being a single user communication system where no special requirements on the cross correlation properties of the synchronization sequences are necessary. Like in L-DACS1 the synchronization algorithm employed is a conventional Schmidl-Cox algorithm [4]. Therefore, first zeros are inserted between the mapped symbols of the control data as shown in Figure 2. The insertion of zeros in the frequency domain leads to a repetition in the time domain, e.g. if between every symbol one zero is inserted in the frequency domain, the resulting time domain signal consists of two equivalent signal parts. Thus, after the IFFT the transmit signal has a very useful property: The first OFDM symbol consists of two equal halves while the second symbol consists four equivalent parts. Thus different metrics can be calculated, e.g. a metric comparing the first with the second half of the OFDM symbol. Using those metrics both the starting point of the frame and the frequency offset between the transmitter and receiver can be calculated [4]. For the equalization of the information carried within the synchronization symbols, the channel estimation uses pilots from the adjacent payload block.

The conversion to the time domain by the IFFT is followed by attachment of the CP and the windowing function as described above. The up conversion to the intermediate frequency (IF) of $f_{IF} = 10.7 \text{ MHz}$ employs multiplication with a complex oscillation as well as digital filtering to cope with the problem of aliases. The integrated digital to analog (D/A) converter is followed by a custom RF frontend combined with a power amplifier creating the final transmit signal. This is then transmitted using an omni-directional S-band antenna on the aircraft and a tracking antenna on the ground. The RF frontend's major advantage is that it is dual banded

and it can be configured to transmit in the L-band as well.

Upper Layer and Transmission Protocol

The upper layer complexity in B-DLR may be reduced significantly because, unlike L-DACS1, it is strictly designed as a single user system. Additionally advanced features like the automatic switching between the modulation alphabets or different coding rates according to the channels quality are not implemented. This is due to B-DLR's design as an experimental communication system. For both reasons the upper layer of B-DLR only features a very basic functionality:

- Checking the data integrity for both the control and the payload information. This function is implemented using a CRC. If either one of the CRC fails, the corresponding data is assumed to be corrupted and therefore discarded.
- An automatic repeat request functionality (ARQ) initiating a retransmission in case a packet was lost or corrupted.

Therefore, the control and payload data is assembled as shown in Figure 5.

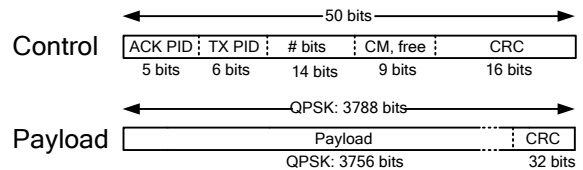


Figure 5. Assembling of control and payload data.

The control channel consists of two packet identifiers (PID) of a length of 5 bits and 6 bits, respectively. The first PID is reserved for acknowledging a previously received packet while the second identifies the packet currently being transmitted. To the transmission PID additionally a new bit identifier is attached, which is toggled every time a packet is transmitted for the first time. If no packet is transmitted, but a packet needs to be acknowledged (ACK), the RX PID is set to all ones, telling no data was transmitted. In case no packet needs to be acknowledged, but a new one is transmitted, the TX PID is set to all ones. The third field is used to signal the receiver the number of bits of payload data in the current packet. Additional

9 bits are reserved to possibly carry coding and modulation (CM) information in a later stage of the development. A CRC of 16 bits assures the integrity of the control data.

To the payload data a CRC of 32 bits is attached. The maximum number of payload bits in one frame differs depending on the modulation alphabet. For QPSK, 3756 bits can be transmitted in each packet.

The automatic repeat request functionality is implemented using a fixed number of transmit and receive ARQ-buffers each containing one packet. Each buffer is directly connected to a PID and is flagged with the number n_{rt} representing the transmission attempts for that PID. Whenever an acknowledge for a certain PID is received n_{rt} is set to all ones, denoting the packet has been received correctly. The transmitter consecutively checks one buffer in each transmission cycle in an increasing order.

- If n_{rt} is all ones, a new packet is written into the buffer and directly transmitted in the same cycle. n_{rt} is reset to zero denoting that this is the first transmission attempt. The new data bit is toggled.
- For any other attempt n_{rt} is increased and the content of the buffer is retransmitted.
- For n_{rt} an upper limit applies. This is necessary to avoid running into a locked

state in case of the communication link breaking down. Thus, if that limit is reached, an error message is issued.

The ARQ buffers are followed by a reordering buffer assuring the packets are output in the correct order.

Implementation Details

In the following section details regarding the implementation of B-DLR as a real time communication system are given. The system is partly based on the L-DACS1 demonstrator developed by the DLR [5]. The section starts with an overview of the demonstrator's hardware and software components. In the second part the implementation of the algorithms in the field programmable gate array (FPGA) technology is briefly discussed.

The system consists of three major parts: an FPGA based signal processing platform running a server application, a client computer running the client application, and a RF frontend converting the IF signal from the signal processing platform to an RF signal and vice versa. An additional power amplifier may be added after the RF frontend. The test setup which is depicted in Figure 6 consists of a ground station (GS) and an airborne station (AS) whereas GS and AS consist of the three parts mentioned before.

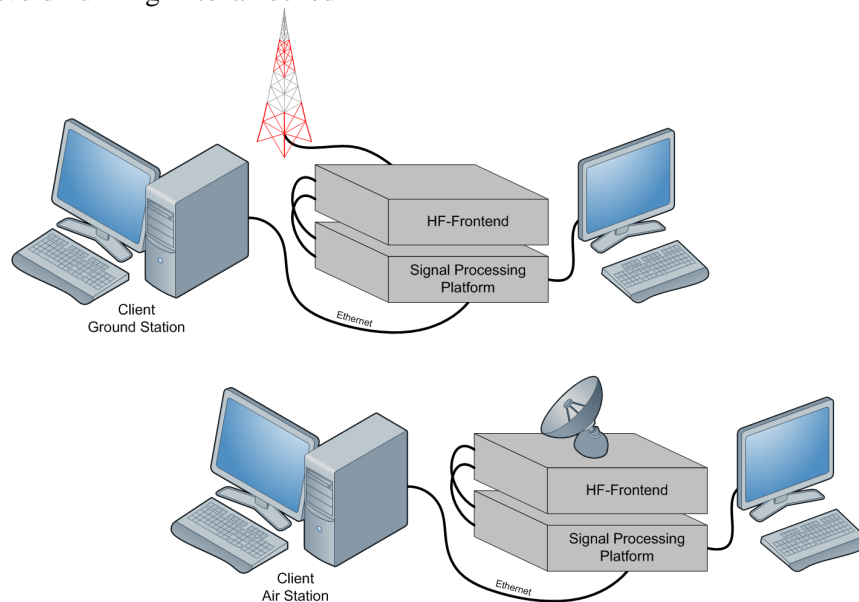


Figure 6. B-DLR demonstrator overview.

The signal processing platform was developed by Parsec [6] and is commonly used in telecom or radar applications. The configuration of the B-DLR demonstrator, shown in Figure 7, consists of a standard cPCI (compact PCI) chassis which is equipped with a single board computer (SBC), two PM410 carrier modules which carry two PMC modules each (one A/D or D/A module and one FPGA processing module), and two PM420 rear transition modules. The SBC also hosts a PM440 clock & trigger module which provides the sampling clock to the A/D and D/A module. The SBC is further equipped with a network interface card (NIC) which is used to feed the signal processing platform with data to send and to transfer the received data back to the client. The SBC runs the Microsoft Windows XP Professional OS (not XP Embedded)

and Parsec provides drivers to access the PM4xx modules from software. Because standard Windows XP Pro is used developing an application for the SBC is straight forward.

Due to the low processing power of the SBC's mobile processor all higher level tasks, e.g. pre-/post-processing, protocol, client application, are sourced out to another computer, the client. For communication via the network interface Windows sockets are used. This allows running both the server and client application on the SBC by using the local loopback (127.0.0.1) in case it turns out that the SBC's processing power is sufficient and no pre- or post-processing in software is necessary. However the functionality to process the data from and to the FPGA in Matlab offers valuable possibilities while debugging or implementing new features.

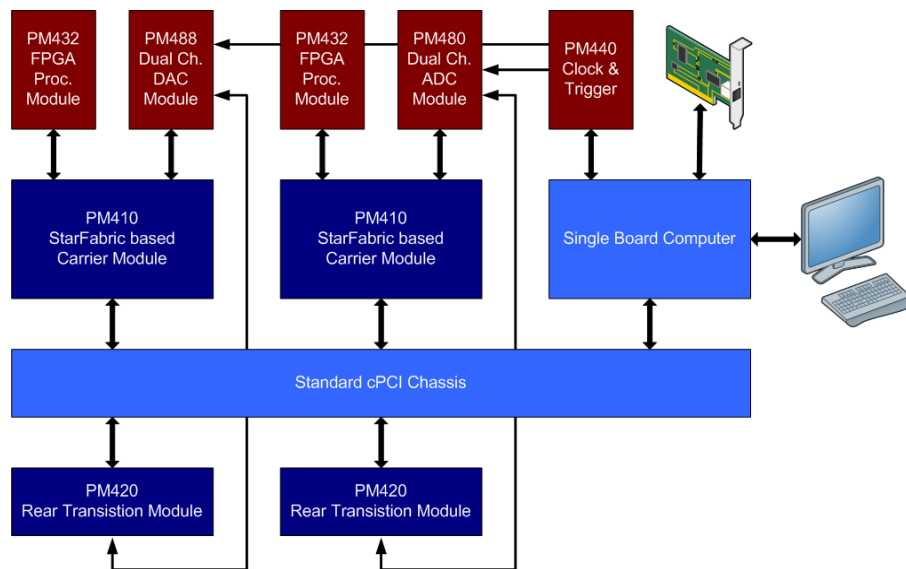


Figure 7. Parsec signal processing platform.

The client application as it is implemented now, shown in Figure 8, makes use of two MATLAB engines for pre- and post-processing, in this case the upper layer functionality. MATLAB provides a C application programming interface (C-API) which allows us to start a MATLAB instance, and use it as a co-processor. The MATLAB API provides several functions for transferring variables from C to the MATLAB workspace, from the MATLAB workspace to C application as well as to execute commands in MATLAB. For not having to recompile the C application each time the commands to be

executed are changed, simply a MATLAB script has to be called from the C application. Thus if the steps for pre- or post-processing need to be modified, it suffices to change a MATLAB script instead of recompiling the software.

For sending data, the client generates the user message, i.e. a sequence of bytes or characters and the length of the message, and adds it to a buffer. By adding the message to the buffer an event is generated waking up a worker thread which gets the message from the buffer and transfers it to the MATLAB engine's workspace. After pre-processing the worker thread gets the pre-processed message

from the MATLAB engine's workspace and adds it to another buffer. Adding the pre-processed message to the buffer also generates an event which wakes up another worker thread sending the message to the

server application running on the signal processing platform via an Ethernet connection. The steps for receiving a message by the client are similar, but the order of steps is reversed.

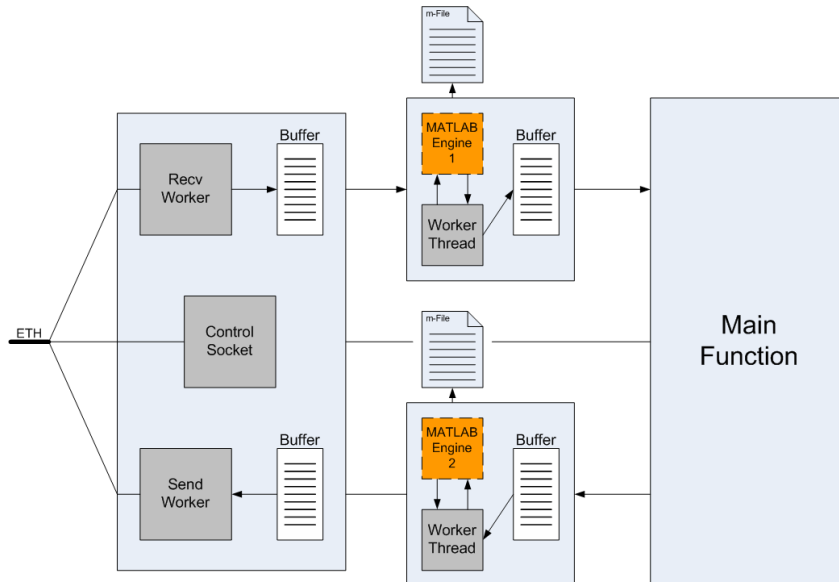


Figure 8. Client application.

The task of the server application runs on the SBC as shown in Figure 9. Its task is to forward received packets to the transmitter design, which is programmed into one of the PM432 FPGA processing modules, and to send the packets from the receiver design, programmed into the other PM432 FPGA processing module, via Ethernet to the client. To achieve full duplex operation, separate tasks for reception and transmission have to be implemented. This is done by using the PMC module's interrupt requests (IRQ). When an IRQ occurs a callback function for the IRQ is executed which has to handle the IRQ appropriately. The server application is also responsible for configuring the PMC modules and setting up the DMA transfers. Once the system is initialized receive and transmit tasks operate asynchronously (not blocking each other) allowing full duplex operation.

Receiver/Transmitter FPGA design

As mentioned before the receiver's and transmitter's physical layer (PHY) is implemented in digital logic inside an FPGA, using well known signal processing techniques. A major advantage of using an FPGA

instead of a digital signal processor (DSP) is, that an FPGA contains a high number of so-called "DSP blocks" which are equivalent to a DSP's MAC (multiply & accumulate) unit. Those DSP blocks can operate truly parallel making the execution of signal processing algorithms, e.g. a FIR filter or an FFT, possible in real-time, which is not guaranteed in DSP technology.

The receiver/transmitter design is implemented on the PM432 FPGA processing module. The PM432 is connected to the Parsec signal processing platform via the PCI bus. For implementing the PCI interface on the PM432 an Altera Stratix EP1S10 FPGA is used. The required PCI firmware is developed and factory programmed by Parsec. The user firmware is implemented in a separate FPGA on the PM432. The user FPGA, an Altera Stratix II GX EP2SGX90, is connected to the PCI FPGA via a proprietary local bus.

Parsec provides a reference design for the user FPGA showing how to implement registers which can be accessed via the PCI interface and also how to implement an Avalon-Streaming interface which is commonly used by Altera's signal processing intellectual property (IP). This reference design was

used as a starting point for the receiver or transmitter design.

All physical layer algorithms for the receiver and transmitter are fully implemented in FPGA. The software used is Altera’s graphical design tool DSP

Builder which is an add-on for MATLAB Simulink. Using this graphical tool increases productivity and allows simulation of the design in every design step without having to write a dedicated test bench just by using the block provided by Simulink.

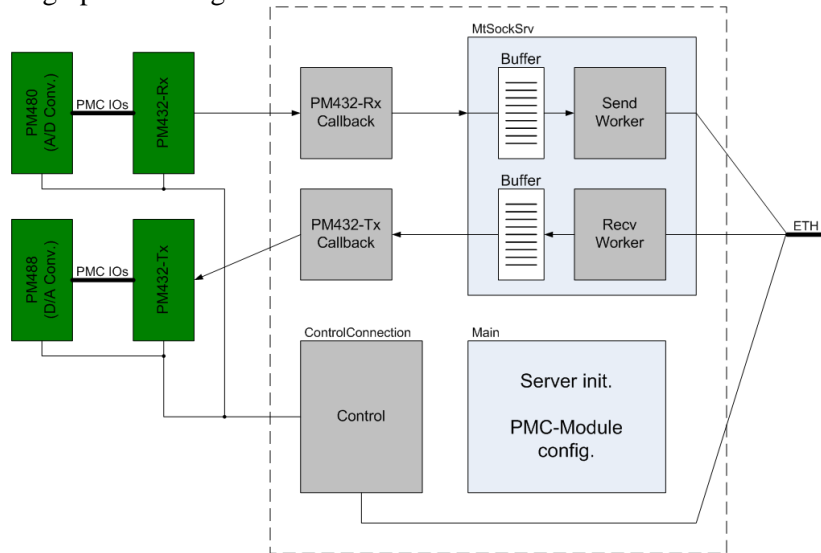


Figure 9. Server application.

Performance Estimation

As for any communication system, and especially in the aeronautical environment, the coverage range is of great importance. Therefore in this section results from simulations are shown, giving information about which carrier to noise ratio (C/N) is needed in order to decode the transmitted data correctly. With that information a link budget can be set up offering a first impression about the range of operation of B-DLR. The link budget partly follows the assumptions made in the L-DACS1 standard [2]. The section closes with a short estimation of the system performance in terms of data rate.

As stated above, two different coding schemes are used for B-DLR: For the payload data a concatenation of an outer RS code and an inner convolutional code is employed, leading to an overall coding rate of $r_p \approx 0.45$. The control data is coded at a lower code rate of $r_c \approx 0.08$, leading to a significant reduction of the energy per carrier needed for a correct decoding. Therefore the overall systems performance is only dependent on the coding of the

payload data. Results of the concatenated RS-convolutional coding scheme are shown in Figure 10.

As a desired packet error rate $PER=10^{-3}$ is assumed. Therefore the necessary C/N is fixed to roughly 2.5dB. Using that value the link budget can be set up. In Table 1 the link budget is shown for the transmission from aircraft to base station, however the resulting range of 68 nautical miles (nm) is valid for both directions.

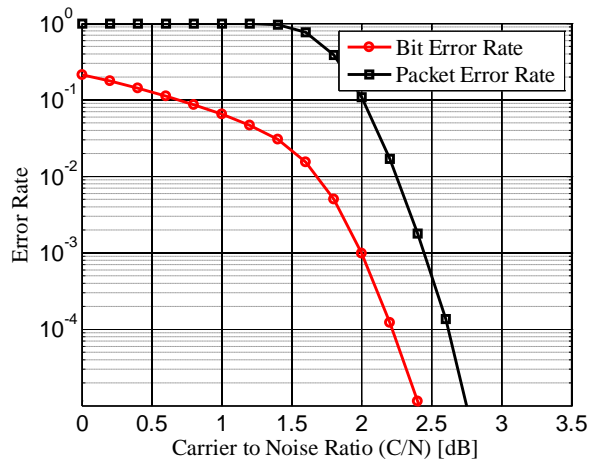


Figure 10. Error rates for payload data.

For the link budget the following assumptions are made:

- For the omni directional airborne antenna a rather conservative gain of 0 dBi is assumed. However that figure already includes a possible banking loss.
- The cable losses at both the airborne and the base station are expected to be not more than 2 dB. This value may be reached by using high quality cable with a loss of 0.05 dB/m assuming a maximum cable length of 40 m.
- At the ground station a tracking antenna with an assumed gain of 12 dBi is employed.
- The noise figure for both airborne and ground station's RF frontend is given with 5.5 dB.
- An implementation margin of 4 dB is assumed. This rather pessimistic assumption is taken due to the experimental character of the entire system.

Table 1. Link Budget for B-DLR.

TX Power	39	dBm	
TX Cable Loss	2	dB	
TX Antenna Gain	0	dBi	
Free Space Loss @ 68nm	142	dB	
RX Antenna Gain	12	dBi	
RX Cable Loss	2	dB	
<i>RX Power</i>		-95	<i>dBm</i>
Noise Density	-174	dBm/Hz	
Bandwidth	66	dBHz	
RX Noise Figure	6.5	dB	
Implementation Margin	4	dB	
<i>Total Noise Power</i>		-97.5	<i>dBm</i>
Required C/N	2.5	dB	
<i>RX Sensitivity</i>		-95	<i>dBm</i>

With the parameters described above each B-DLR frame can carry a payload of 3756 bits, if QPSK modulation is employed. The duration of one frame is fixed to $t_{\text{frame}} = 1700 \mu\text{s}$, thus the overall expected payload data rate is about 2.2 Mbit/s. The resulting bandwidth efficiency is roughly 0.5 bits/Hz/s. This

bandwidth efficiency is relatively low compared to that of today's wireless systems. This is mainly due to two reasons:

First the QPSK modulation alphabet is combined with a forward error control coding of a rate of less than a half. Hence combined with the forward error correction the bandwidth efficiency is already reduced to less than 1 bit/Hz/s. However if first tests provide promising results, both the modulation order and coding rate can be increased to higher values.

The second reason for the low bandwidth efficiency is the large synchronization and AGC symbol overhead compared to the relatively small number of payload OFDM symbols. A possible solution is to increase the number of payload OFDM symbols in a frame. This however makes the segmentation of the payload data into smaller code blocks necessary in order to keep the packet error rate low. The resulting increase in terms of implementational complexity, especially the need for a more sophisticated upper layer protocol is currently not foreseen for the experimental radio link B-DLR.

Conclusion and Outlook

With the implementation of B-DLR a high bandwidth radio link for ATRA is developed. Due to its close relationship to L-DACS1 also a simplified demonstrator of L-DACS1 is available. The functionality of the frontend to transmit both in the L- and S-band enables flight tests for L-DACS1's fitness as the new L-band communication standard.

The work on the transmitter was finished beginning of 2011. Currently the last receiver blocks are being implemented in FPGA. First tests for the new communication system are planned for early summer 2011. If these tests give promising results the next step will be the certification of the equipment in use for flight tests. These tests will then first be performed on a smaller aircraft prior to the implementation on ATRA.

References

- [1] S. Brandes, U. Epple, S. Gligorevic, M. Schnell, B. Haindl, M. Sajatovic, May 2009, "Physical Layer Specification of the L-band Digital Aeronautical Communications System (L-DACS1)", *Integrated*

Communications Navigation and Surveillance Conference (ICNS 2009), Arlington, VA, USA.

[2] M. Sajatovic, B. Haindl, M. Ehammer, Th. Gräupl, M. Schnell, U. Epple, S. Brandes, February 2009, “L-DACS1 System Definition Proposal: Deliverable D2”, *Eurocontrol Study Report*, Edition 1.0.

[3] www.sesarju.eu

[4] T. Schmidl, D. Cox, “Robust frequency and timing synchronization for OFDM”, in *Transactions on Communications*, Vol. 45, pp 1613-1621, 1997

[5] N. Franzen, A. Arkhipov, M. Schnell, “L-DACS1 Physical Layer Laboratory Demonstrator”, in *Proceedings Integrated Communications Navigation and Surveillance Conference (ICNS 2010)*, Herndon, VA, USA, May 2010

[6] www.parsec.co.za

E-Mail Adresses

Nicolas.Schneckenburger@dlr.de

Christoph.Klein@dlr.de

Michael.Schnell@dlr.de