Adaptative integration systems using FPGA COTS devices.

Elói Fonseca; André Pierre Mattei*; Wagner Chiepa Cunha**
** Instituto Tecnológico de Aeronáutica, ITA*
Praça Marechal Eduardo Gomes, 50, São José dos Campos, SP, Brasil

Abstract
This work proposes a conceptual structure of systems integration using Field Programmable Gate Array (FPGA) and lookup tables to define an adaptive integration structure, providing a communication link between systems that don't work with the same protocol. An implementation using Commercial of The Shelf (COTS) transceivers as front-end in addition with FPGA devices for the processing of the message code, where VHSIC Hardware Description Language (VHDL) routines can be organized in blocks of encoders/decoders, in this way providing logical resources to translate protocols. Additionally, through the use of lookup tables with these FPGA, the capability of elaborating more complex algorithms can be acquired. The aim of this paper is to model an adaptive structure of integration systems, using FPGA devices and lookup tables. Experiments with applications for UAVs and CubeSats were developed in the Integration Systems Laboratory from the Instituto Tecnológico de Aeronáutica (ITA), using COTS transceivers, COTS FPGAs and software implementation tools from XILINX.

1. Introduction

Traditionally, the standard buses that can be found in communication and control systems [1] are RS-232, RS-422, RS-485, USB or Ethernet with protocols such as DeviceNet, CANopen, Profibus, Transfer Control Protocol/Internet Protocol (TCP-IP), Synchronous Data Link Control (SDLC), High-Level Data Link Control (HDLC) or SpaceWire. This variety is increasing, requiring interfaces to cross a wide range of applications, such as Communication Systems, Programmable Logic Controllers (PLCs), Input/Outuput modules, motors, sensors, etc. Military, industrial and medical networking protocols need to establish modular communication, allowing integration between components from different manufacturers. Figure 1 shows a process of communications extended for a multiple point network, that can be classified into six levels:

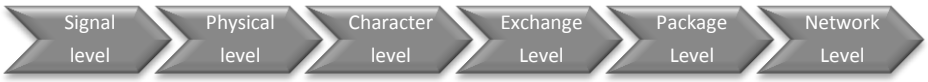

Figure 1: Structure of a communication device in a system network. Adapted from [2].

In order to provide a ready-to-use development platform for system designers to quickly and easily implement programmable systems integration for applications in which the main system needs to control different data streams, such as logical or physical protocols (e.g. RS-232C, RS-422, RS-485, USB, SpaceWire, ETHERNET, TCP-IP, SDLC, HDLC, CANBUS, ARINC429, MIL-STD-1553), it is useful to work with some resources as FPGA devices that can offer flexibility to support changing standards, parallel processing and customizable interfaces with integrated functions, like high performance DSP, memory, analog and I/O interfaces, making it easier for designers to embed soft processors to control dataflow and manage a number of interfaces in the system. The knowledge

Copyright © 2013 by Elói Fonseca; André Pierre Mattei; Wagner Chiepa Cunha. Published by the EUCASS association with permission.

obtained in message management and data frame interface used in Radio Defined by Software (RDS) systems was applied in this work to define a basic algorithm structure.
This capability in integrated systems with low mass and electrical power is very useful in UAVs and CubeSats, where devices from several suppliers need be used in a fusion data system for the mission. In this research, COTS devices are used in order to allow the application in academic projects and the analysis of flight performance of these in embedded Real Time Systems architectures. Providing fast and efficient communication links for peripherals, it is possible to achieve reliability and fast response in critical events.

2. Requirements of systems integration using FPGA

When two or more systems need to be integrated, several specifications must be fulfilled. Firstly, at the signal level, it is possible to observe requirements for voltage magnitudes, impedance networks and coupling rules. Table 1 shows basic physical requirements for specific patterns.

Table 1: Physical levels for the most frequent communication protocols

	Level	**Line**	**Coupling**
RS-232C	±12VDC	Unipolar	Drivers
RS-422	+5VDC	Differential	Drivers
USB	+5VDC	Unipolar	Directly
ARINC 429	±10Vp	Differential	Transformer
MIL-STD-1553B	±18-27Vp	Differential	Transformer
SpaceWire	+5VDC	Differential	Drivers
CANopen	+12VDC[a]	Unipolar	Drivers

[a] The CAN protocol presents specific rules for level in each application.

When a system needs to be connected to another, a coupling driver that is compatible with the impedance must be used, as well as levels and security options included in it. Some COTS today have enough temperature, vibration and electromagnetic compatibility requirements for their use in UAVs, aircrafts and inclusion in CubeSat projects.
Physically, some responsibilities as encoding/decoding and signalling functions manage the data flux from/to the device allowing receiving and transmitting data between these systems as showed in the Figure 2.

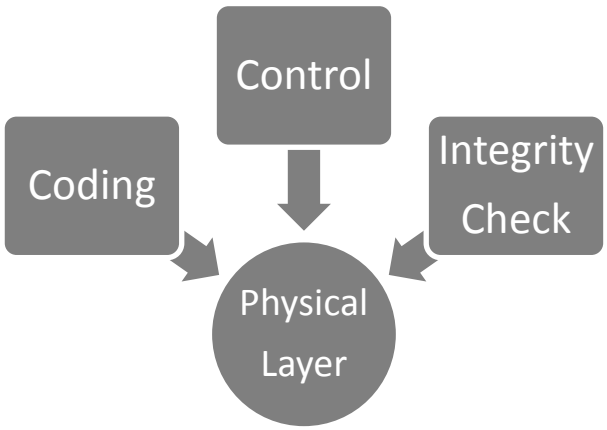

Figure **2**: Structure of a Physical layer function. Adapted from [3].

Each protocol has codes and controls specifics and when one system project needs to integrate different technologies, the most common solution is to perform each integration using individual interfaces with ASIC devices and including an Operational System specific for this system project.
Another line of development is used by Aeroflex Gaisler, where a IP core for implement a LEON 3 processor is used for several COTS FPGA devices, in a GPL distribution, and works with several communication protocols included in the decryption code, as showed in the Figure 3.

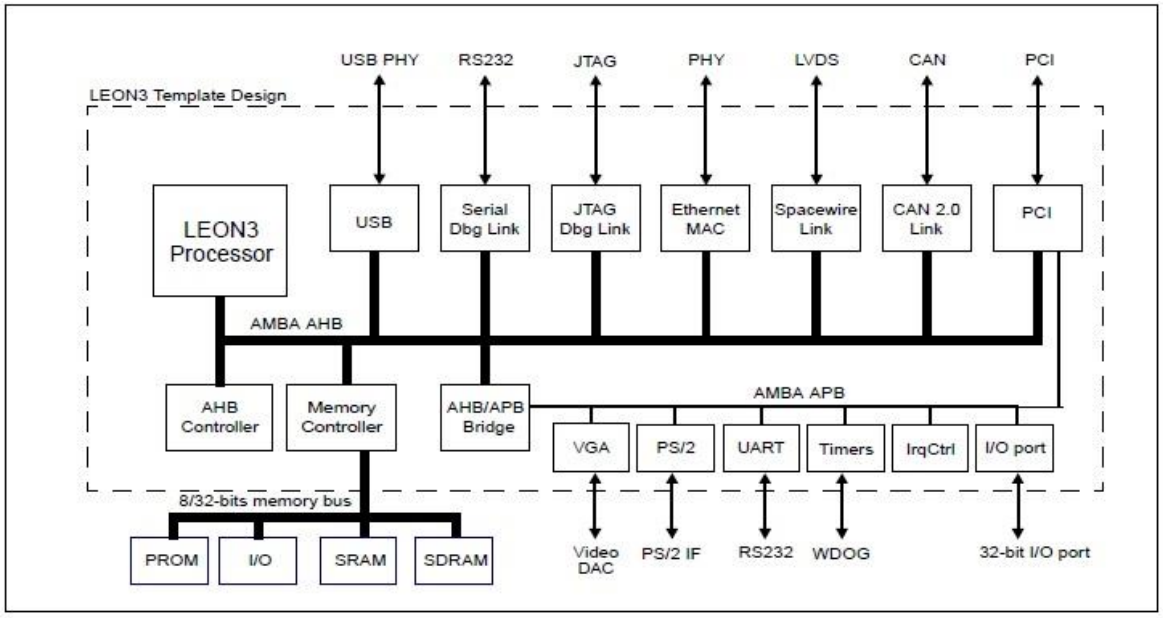

Figure **3**: Structure of a Leon 3 processor IP core for VHDL [4].

The main problem remains in a characteristic that is common to all applications of integrated systems: Each project requires a specific set of settings for each specific device data pattern (code, frame composition, data bock, etc.), as a matter of fact, if for each new project it is created a new structure of settings sets, that is too complex to do so fast.
If possible to use a modular structure, where flexible interfaces described in VHDL blocks, where the settings are done by parameters that can be reached in lockup tables included in the VHDL system description, it is possible to have a adaptive system, in the concept named Mission Oriented System, where the system can be adaptive for several applications, only changing the configuration data instead to change the full code.
The concept of package VHDL blocks in FPGA project for interfacing several protocols, organized in a library structure, as show in the Figure 4, improve this capability in scalable modular complexity architecture, looking for development an adaptive system with multi-protocol operation capability.

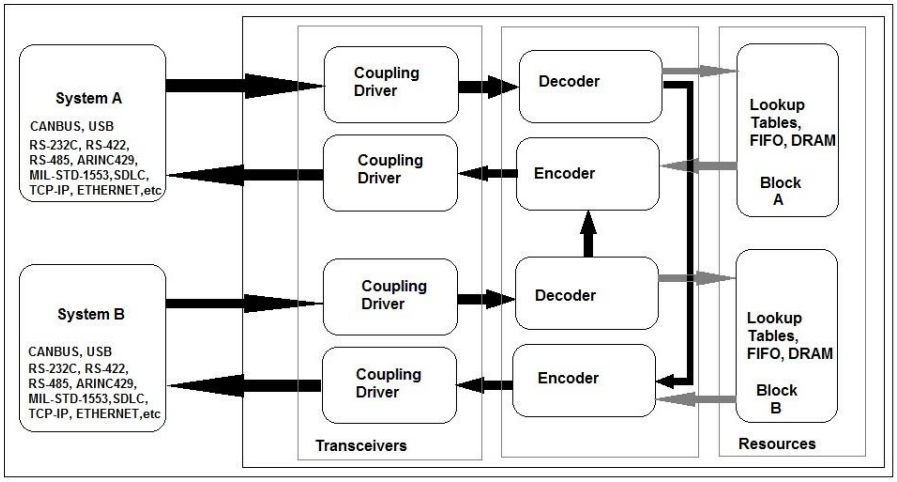

Figure **4**: Structure of a multiprotocol interfacing system using lock up table blocks.

This structure can be easily replicated in VHDL description in multiple ports for as many external devices as needed, forming a modular construction, whenever the libraries allow it. The blocks for encoder/decoder functions, controls and errors check, and all of additional logical resources can be implemented in the FPGA by VHDL description, and the coupling drivers are done external to the FPGA by COTS devices.

3. Experimental development

In order to build an experimental model, some tests for development of integration systems in a FPGA device were started with some specific protocols, developing libraries for each one in blocks can be called by a description code. The optimization of the code in these libraries is essential to improve the capability to use as a tool that allows a modular implementation of complex system using them.
The main idea was that it is a better way to develop some library blocks for specific protocols than performing experiments with replication more than one port using these libraries and check the capability in use then in several projects with the same libraries blocks developed.
The first integration test executed with an inertial device had a communication line for RS-485 protocol, and logical message using SDLC protocol. This device uses a complex data frame and needs a dedicated algorithm to receive and decode the message frame.
In this experiment, COTS RS-485 line drivers DS96174 and line receivers DS96175 from Texas Instruments were used as analog front-ends of the FPGA device, and a description using a SDLC decoder, that has included an asynchronous decoder, was built, performing a serial/parallel converter with reverse stuffing correction algorithm included and after that a frame decoder with a counter block and control for FIFO transfer resources to allow to keep the messages decoded for test. The parameters for the data decoder and frame decoder are send by Look-up Tables (LUT) for this purpose.
The code optimized have been used in simulation by software and after that were developed a SDLC frame block generator in VHDL description, allowing phase tests without the need of a real device for generating frame data.
The Xilinx Spartan 6 FPGA devices used in the SP601G development board were programmed using the ISE 13.1 software development application supplied by the manufactory XILINX. This experiment block description is displayed in Figure 5.

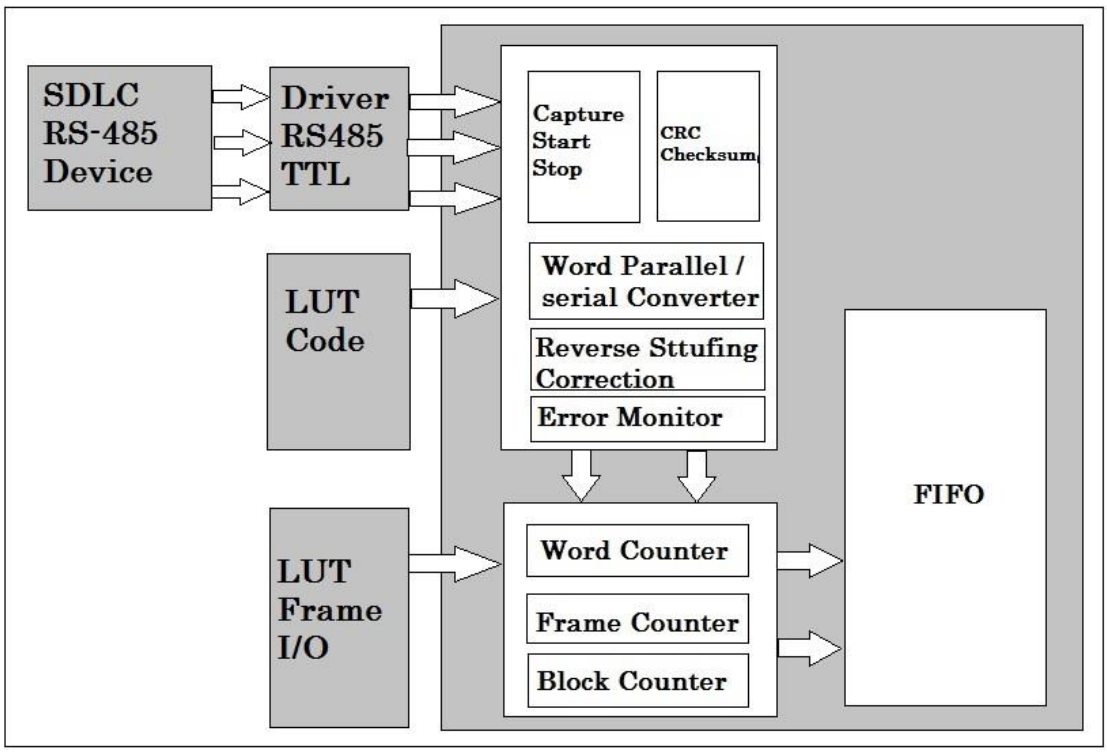

Figure 5: Structure of experimental development of a SDLC frame decoder.

These experiments was proved using a known message from the device, using a VHDL code for simulated this pattern and some results of these experiments can be seemed in the Figure 6.

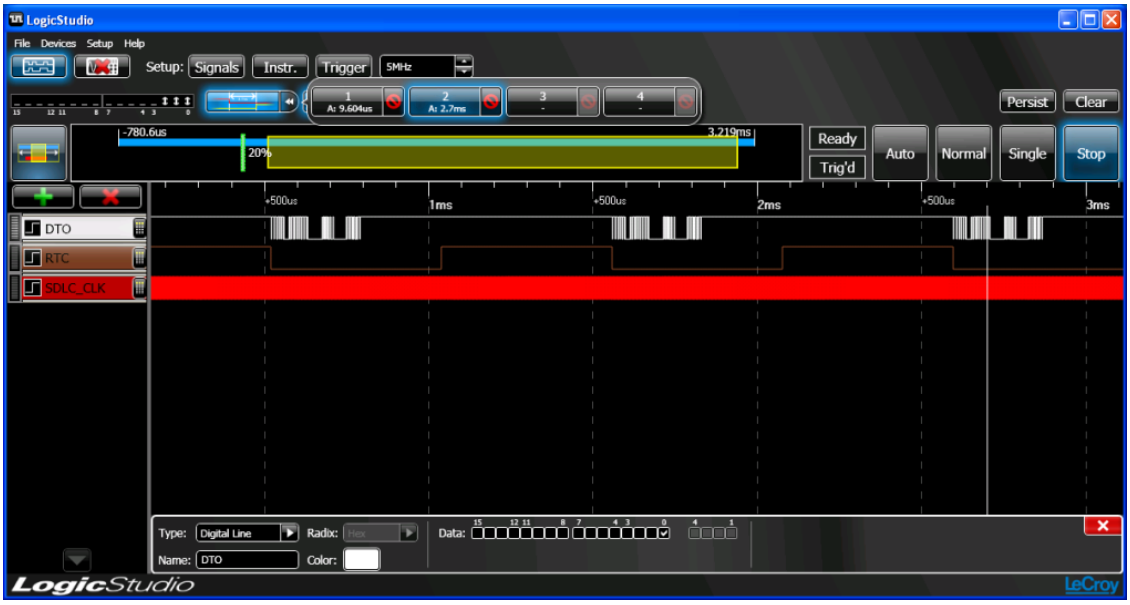

Figure 6 – Test of a SDLC frame decoder using a LeCroy USB Logic Analyzer.

Another experiment was a Holux GPS integration, using a RS-232C protocol with a frame data in 4800bps. For this project, a MAX 232C line driver was used as a front-end and a serial port was created in a VHDL description, with the frame decoder for the GPS message included and a FIFO implemented to allow a controlled data transfer for another application device by another serial port, that were replicated by a call for the same block library for a serial port using a external modem in 9600 bps and these data were received by another computer by a RF modem as can be seemed in the Figure 7.

Figure 7 – Serial port RS-232C experiment, with dual port included.

This experiment proves the capability in receive and decode a frame message that contain several data blocks, register this message in a FIFO memory block forming a database and after that to choose some specified data from this database and create another frame message that was send by another port created in the VHDL code, using the same description of a serial block library, but using another settings form a Look up table.
The message was monitored by another computer, as described, and the signals in the implemented setup were proved using one LeCroy USB Logic Analyser, both serial port were monitored and the system worked as planned, the Figure 8 shows these results.

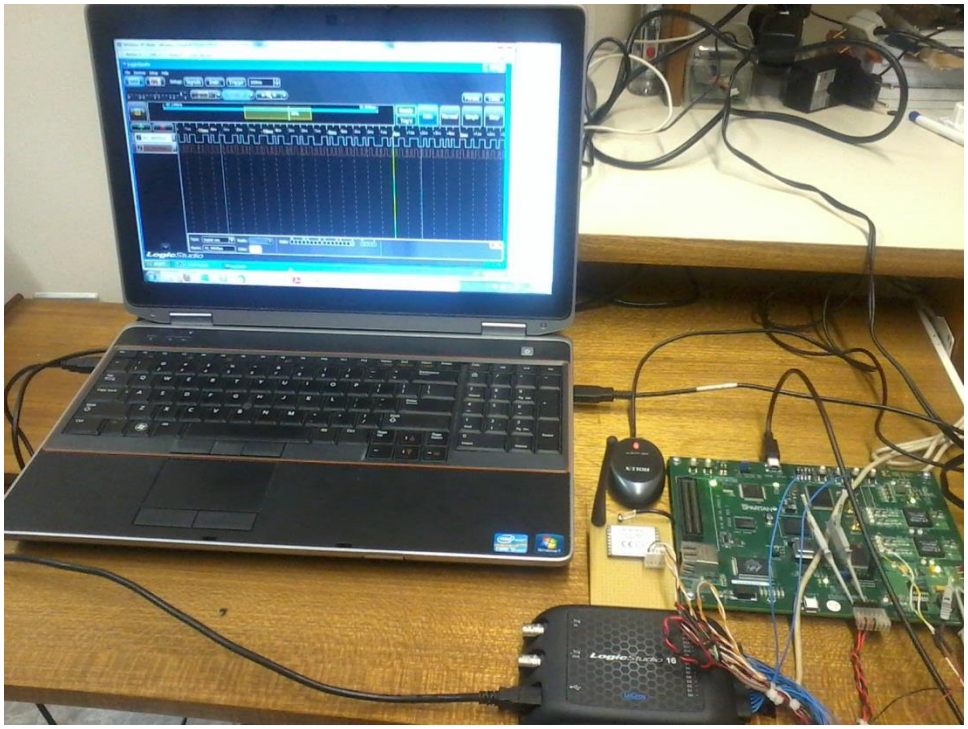

Figure 8 – Setup for the Holux GPS integration test.

4. Results and comments

These experiments allowed checking the practical optimization of VHDL blocks, where core generators tools can be used, but after the initial generation an optimization process for generate generic application blocks, changing some details and refining the code structure of these models. After this work these blocks can be joined in package VHDL libraries for use in system projects.

It is not so easily, because the core generator tool create a complex code, with several resources that can be not so useful for embedded systems, in this case the main work is identify these resources, implications in all of code body and extract only what is possible.

Using look up tables specifics for code settings and frame settings was possible to demonstrated that the same block library can be used in several ports for distinctive code and frame patterns. This was a good way to improve a capability to generate faster new integration projects reusing block libraries done in another work.

The capability in use COTS FPGA for UAVs in interfacing, communication and control systems done to use another board, specific for prototype systems, but with a generic open design, the board choose was the OPALKELLY XEM 6001, that was used in the same tests done with the SP-601G and worked as planned. This board is showed in the figure 9.

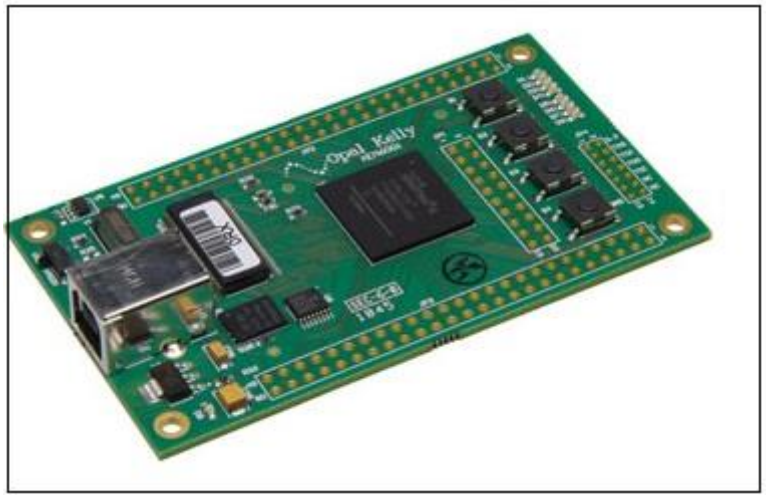

Figure 9 – Prototyping Board XEM 6001[5].

With these board, that has low mass and power budge, and friendly connectors that can be used with a flat cable, its possible expand embedded applications targets, as UAVs, Drones, aircrafts or small experimental satellites as cubesats.

5. Future works

After the experiments and results obtained, the structured library was implemented, using generic blocks for each logic protocol and organized in VHDL packages can be replicated as the project need[6]. The configuration settings for use with the look-up tables [7] allowing to change the control pattern for each device are been implemented in this research phase, for a multiple mission unmanned terrestrial vehicle, that uses a GY-85 MEMS IMU 9DoF (Degrees of Freedom) with a I2C protocol, a GPS SKYLAB SK-53M, with a serial port RS-232C as a protocol and a RF transceiver RF24L01, 2.4GHz, 2 Mbps that works with a ISP protocol. This experiment is showed in the Figure 10.

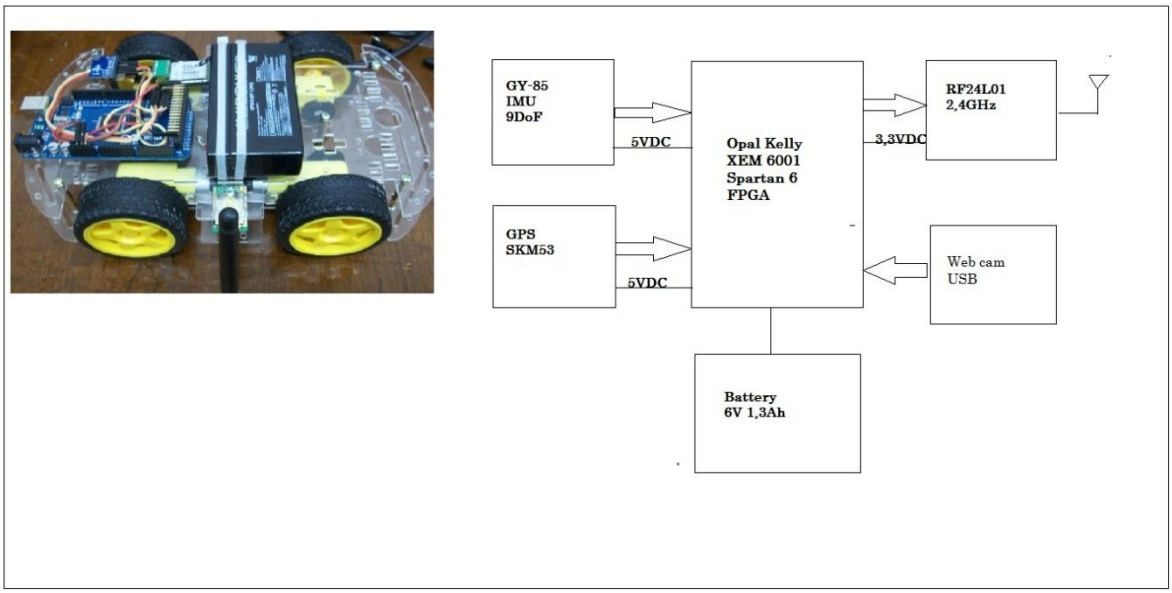

Figure 10 – System integration of an unmanned vehicle.

The objective of this experiment is to prove the capability in integrated data from sensors that work with specific protocols, doing the interfacing for these devices, achieving they data frames, and including making a data fusion for INS/GPS navigation, all of these functions implemented in a FPGA using structured VHDL libraries as seemed in this research. The functions and devices in this project will be changed several times in order to prove the practical

utility of the structure using the same libraries set and look up tables for configure the devices data integration as needed in each set of devices allocated.
In this experiment will be used a modular structure proposed in this work, with a inclusion of a video camera USB to prove some image processing algorithms for image tracking and gyrostabilizer camera control in this embedded system.
The set of device options for this experiment is showed in the Table 2.

Table 2: Some devices used in the experiment 3.

	Function	**Protocol**
Holux GPS	GPS receiver	RS-232C
SKM53	GPS receiver	RS-232C
GY-80	IMU	I2C
GY-85	IMU	I2C
GY-87	IMU	I2C
SD card	SD card slot	ISP
WEBCAM	Image sensor	USB

Another experiment will be done using a MIL-STD-1553B protocol in a VHDL block and another ARINC 429 VHDL block, the front-end with COTS transceivers will be used to coupling these signals. Using these two patterns, will be implemented a data network for simulate an aircraft navigation complex systems, using one call for each port that will be created, and using the look-up tables to configure each device that will be simulated. This research will be done with the resources from a Embedded Systems laboratory in the Instituto Tecnológico de Aeronáutica, that has a network with several computers working with commercial boards for these protocols.

5. Conclusions

The use of a adaptive modular model for systems development is a way that improve capability in work with several protocols, in a same project line, but for each mission configuration it's needed only change the parameters of the configuration, but not change part of the structural project.
For UAVs or satellites this capability can be useful to allow simplify qualifying process for flight models because the operational system is pattern, and the resources are tested for qualification. Minimal changes are needed for new applications if needed the same chart of protocols. In these projects a optimized structure reflects a lower power budge [8], improving a tool to reduce a power problem in critical projects as cubesats and UAVs.
Similarly of the Leon 3 GRLIB, a GPL distributions could be a tool for another contributions and collaborate in several projects for UAVs and cubesats.

Glossary

ARINC 429 – Communication Protocol
COTS – Commercial Of the Shelf
FPGA – Field Programmable Gate Array
ETHERNET – Network communication protocol
GPL – General Public License
GPS – Geostationary Position System

GRLIB – Library from Aeroflex-Gaisler
HDLC - High-level Data Link Control
LEON 3 – SPARK Processor (used in ESA projects)
MIL-STD-1553B – Communication Protocol
RDS – Radio Defined by Software
RS-232C – Serial Communication Pattern
RS-422 – Serial Communication Pattern
RS-485 – Serial Communication Pattern
SDLC - Syncronous Data Link Control
TCP-IP – Transfer Control Protocol – Internet Protocol
UAV – Unmaned Aerial Vehicles
USB – Universal serial Bus
VHSIC - Very High Speed Integrated Circuits
VHDL – VHSIC Hardware Description Language
XILINX – Commercial FPGA Manufactory

References

[1] P. A. Laplante, " Real-Time Systems Design and Analysis," in IEEE Press 2004.

[2] J. Bard; V. J. Kovarik Jr, " Software Defined RadioThe Software Communications Architecture," John Wiley & Sons ltda, 2007.

[3] Proakis, J. G. “Spread Spectrum Signals for Digital Communications”. Encyclopedia of Telecommunications. 2003

[4] Aeroflex Gaisler “GRLIB IP Library User’s Manual”. 2013 http://gaisler.com/products/grlib/grip.pdf. [Accessed 10 May 2013].

[5] OPALLKELLY “XEM6001 User’s Manual”. 2012 http://assets00.opalkelly.com/library/XEM6001-UM.pdf [Accessed 10 May 2013].

[6] d’Amore R.. “VHDL Descrição e Síntese de Circuitos Digitais”. LTC, 2012 - ISBN 978-85-216-2054-9

[7] T. Sasao and J. T. Butler, “A Design Method for Look-up Table Type FPGA by Pseudo-Kronecker Expansion,” Proceedings of the 24th International Symposium On Multi-Valued Logic, 1994, pp. 97-106.

[8] F. Li, D. Chen, L. He, J. Cong, "Architecture Evaluation for Power-Efficient FPGAs," ACM/SIGDA International Symposium on FPGAs, Monterey, CA, 2003, pp.175-184