AUTOMATYKA/ AUTOMATICS • 2013 • Vol. 17 • No. 1

http://dx.doi.org/10.7494/automat.2013.17.1.93

Sławomir Samolej*, Tomasz Rogalski**, Grzegorz Kopecki**, Andrzej Tomczyk**

The Integration of a Prototype Pitch Control Application with IMA2G Devices

1. Introduction

A modern airliner's computer system include a wide set of hard real-time applications. Typical avionic components such as cruise speed, flight level or pitch angle control systems are good examples. It is worth remarking that the avionic hard real-time systems [4, 5] usually have to both execute control tasks and exchange data between each other as well as the pilot's cockpit.

One of the main objectives of modern aviation industry is the development and implementation of Integrated Modular Avionics (IMA) devices onboard of airliners. Following the IMA concept, modern onboard avionic subsystems (software applications) should be grouped in a limited set of standard microprocessor units. The microprocessor units and other electronic devices should communicate via standard Ethernet network interface. The so far physically and logically separated (federated) avionic units must be converted into groups of real-time applications controlled by real-time operating systems. The software modules executed on the hardware units should be developed according to ARINC specification 653 [3, 12]. The Ethernet-based communication interface should follow ARINC specification 664 [1, 2]. The introduction of an IMA concept should bring the reduction of both an aircraft weight and maintenance costs.

The first IMA-based devices have been introduced through European funded research projects PAMELA, NEVADA and VICTORIA and are currently mounted onboard the A380, A400M and B787 aircraft. Research started within the abovementioned programs was continued in SCARLETT (Scalable and Reconfigurable Electronics Platforms and Tools) [8] program where the next generation of IMA devices (IMA2G) was developed. Its main aim was to define a scalable, reconfigurable fault-tolerant driven and secure new avionics platform (IMA2G), called Distributed Modular Electronics (DME).

* Dept. of Computer and Control Engineering, Rzeszow University of Technology, Rzeszow, Poland
** Dept. of Avionics and Control, Rzeszow University of Technology, Rzeszow, Poland

The Rzeszow University of Technology Research Team (RUTRT) has taken part in one of the SCARLETT research paths: time-critical application development and testing. The developed application should have been a part of an aircraft's IMA system and made it possible to evaluate whether DME units could be effectively used as hard real-time applications platforms. Consequently, it was decided that a Pitch Control Application would be the representative hard real-time application. The preliminary research results of RUTRT were published in [18, 19]. Some self testing procedures build into the developed control system and making it possible to assess its quality during the runtime were announced in [16, 17]. The author's experience in ARINC 653 specification based real-time systems development was summarized in [20]. The application was developed for two target real-time operating systems: VxWorks 653 [10, 21, 22] and PikeOS [9, 13, 14].

This paper reports the integration of RUTRT application with IMA2G devices developed within the SCARLETT program. The next sections of the paper are organized as follows, firstly, the ARINC specification 653/664 based distributed real-time systems development and integration rules are introduced. Secondly, the Pitch Control Application developed by the RUTRT as well as its integration with IMA2G devices are presented. The final part of the paper includes some application integration development remarks and RUTRT future research and implementation plans.

2. Standardization documents

Devices constructed in accordance with IMA2G philosophy should be able to effectively exchange data via computer network which follows ARINC 664 specification [1, 2]. Moreover, selected IMA2G devices, especially these responsible for intensive data processing, should have the hardware-software capabilities to execute a set of temporally and spatially isolated real-time applications. The framework for such application development is defined in ARINC 653 specification [3, 12]. Next two subsections present a set of practical development rules of ARINC 653/664 specification based systems.

2.1. ARINC specification 664

ARINC specification 664 is based on IEEE Std 802.3 [11] and defines changes to the basic document only if the aeronautical applications specifications require them. The main aim of the specification is to define the development rules for Ethernet switched networks where it is possible to calculate the transmission delay and jitter between two nodes of the network. To achieve this:

– network devices must be statically configured;
– virtual links are defined to determine network data routes;
– both Bandwidth Allocation Gap – BAG and Max. Jitter are associated to each virtual link;

– switch device scheduling algorithm must preserve the order of data, timing parameters of data as well as two levels of priorities of data;
– data format is strictly defined;
– at least two redundant physical networks should serve the network.

Consequently, each IMA2G device includes build into the configurable table of virtual links. The table shows to which virtual link and from which virtual link network data is transferred. The virtual links are one way connections, where one source of the link may be connected to may destinations. The configuration table also includes the timing parameters of the link: Bandwidth Allocation Gap – BAG and Max. Jitter. Figure 1 shows the interpretation of virtual link timing parameters.

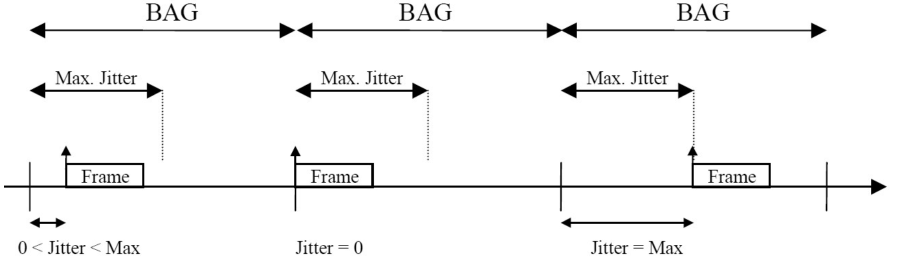

Fig. 1. Interpretation of virtual link timing parameters ([2], pp. 13)

ARINC specification 664 based computer networks transfer data in full duplex mode and are also called the ADFX networks (Avionics Full Duplex Ethernet Network). The BAG parameter defines the "time gap" within which a frame of data should be send. The Max. Jitter parameter informs us what is the longest possible delay of the start of a frame transmission within the BAG. The delay in the transmission may be mainly caused by the switch scheduling algorithm. ARINC specification 664 recommends also the order of data transmitted in a frame. It is possible to strictly define the position of the data and their interpretation at the system specification level. It is also assumed that for safety reasons at least two independent computer networks should redundantly transfer the frames. The end system devices should be able to assess the integrity of the data. If both of the data are integral, the end system should select the frame which came earlier. The network redundancy should assure the proper distributed system behavior even if one of physical network connections is out of order or disturbed.

2.2. ARINC specification 653

One of the main IMA2G concepts assumes the reduction of the number of the individual microprocessor units installed on-board. This entails a new paradigm in avionics development. The group of federated applications which have been executed up to now on separate microprocessor units must become a set of real-time processes executed on one microprocessor. To cope with this level of criticality, new real-time operating system architecture was suggested. The main document defining the generic system structure and its API is ARINC specification 653 [3, 12].

The key concept introduced in the specification is the partition. It constitutes a kind of container for an application and guarantees that the execution of the application is both spatially and temporally isolated. The partitions reside in a separate memory areas (special isolation) and exchange data with the environment by means of specific interface – APEX (APplication/EXecutive). Inter-application (partition) communication is based on a ports and channels concept. The applications do not have the information about which partition the receiver of data is being executed on. They only send and receive data via ports. The ports are virtually connected by channels which are defined in a integration level of system development.

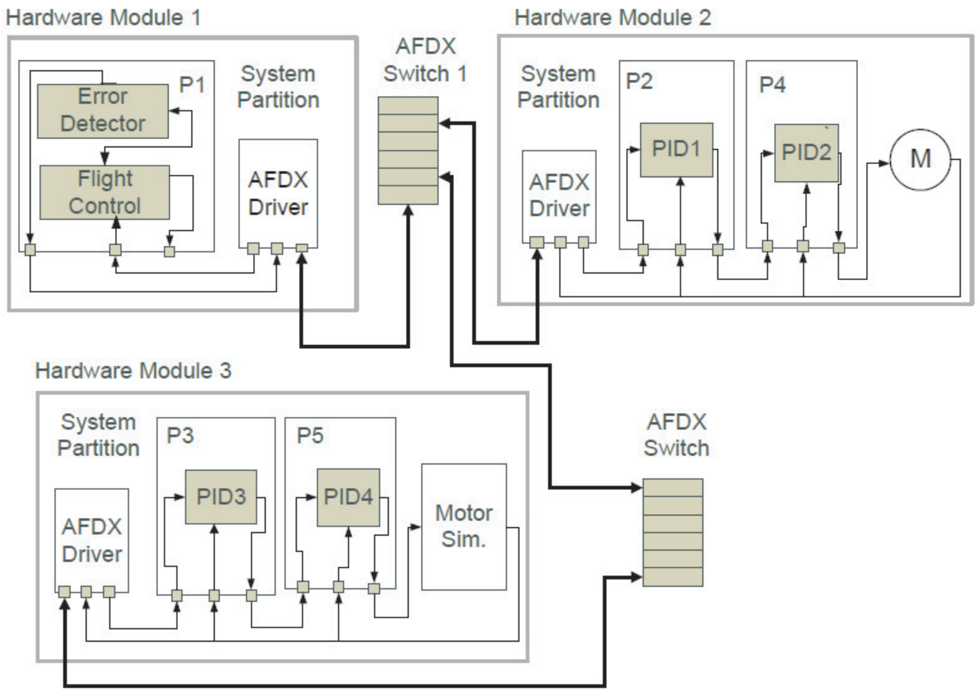

Fig. 2. Hardware-software structure of prototype Pitch Control Application

The temporal isolation of each partition has been defined as follows, a major time frame is defined for each module. It is activated periodically. Each partition receives one or more time partition windows to be executed within this major time frame. Generally, time partition windows constitute a static cyclic executive [4]. Real-time tasks executed within the partition can be scheduled locally according to a priority-based policy. The order of the partition windows is defined in a separate configuration record of the system.

An important element of the module should also be a Health Monitor (HM). It is an operating system component that ought to monitor hardware, the operating system and application faults and failures. Its main task is to isolate faults and prevent failure propagation. As an example, the HM is permitted to restart a partition when it detects an application fault.

Figure 2 includes a hardware-software structure of prototype Pitch Control Application developed in accordance with ARINC specification 664/653. The first hardware module includes two partitions. P1 partition includes two real-time tasks. The second and third hardware modules include three partitions. P2–P5 partitions include one real-time processes. The system partitions are responsible for inter-hardware module communication via AFDX ARINC specification 664 based network. Inter-partition communication is based on ARINC specification 653 ports and channels.

2.3. Integration

Integration of software units (partitions) and the communication structure is a crucial level of IMA2G systems development. In general, the applications that reside on partitions may be developed by separate application providers. Thereafter, a separate role in the IMA2G systems development process called integrator has been proposed. This person or organisation has to collect the data regarding resources, timing constraints, communication ports and exceptions defined in each partition. Then the collected data is transferred into so called configuration records. The configuration record for each module is an XML document interpreted during compilation and consolidation of the software. Simultaneously, separate configuration records are being prepared for hardware module network interfaces and AFDX switches responsible for inter-hardware module data exchange.

According to ARINC specification 653 each hardware module configuration record should include:

- inter-partition communication structure;
- communication channels definition;
- memory requirements for each of the partitions;
- scheduling strategy for the partitions (main and partition time windows)
- health-monitor settings.

The inter-partition communication structure includes ARINC specification 653 based ports and channels. The port names defined in configuration record should correspond with the appropriate port names defined in partition source codes. The ports that belong to the same hardware module can be virtually connected by channels defined in the configuration record. On the other hand, the ports that require inter-hardware module connection are connected to a special system partition (compare Fig. 2) which translates their messages into computer network frames.

A partition memory requirements depends on the code complexity and data structures implemented in it. According to ARINC specification 653 a dynamic memory management is forbidden due to safety requirements.

To configure the partition scheduling the integrator has to define the main time frame and assign the partition time windows to each partition within the main time fame. Each partition time window defines the time slice within which the partition may execute. Additionally, partition windows repetition should be correlated with real-time periodic tasks timing parameters defined within the partitions.

The health-monitor settings decide about the system behavior when the system malfunction is detected. Some level of system damage can be handled at partition application level (compare [17]). System faults that cannot be handled by the partition application are propagated to the health-monitor serving procedures indicated in the configuration record.

According to ARINC specification 664 AFDX network configuration should include:

– virtual link connections between the end systems;
– timing parameters for each virtual link;
– switching tables for each switching device.

Virtual link is identified by UDP port, IP address and MAC of source AFDX port as well as by UDP port, IP address and MAC of destination AFDX port. Simultaneously, timing parameters of each virtual link are defined: Bandwidth Allocation Gap, and Max. Jitter. The Max. Jitter can be calculated using the formula published in [2]. The switching tables defined for the network switches determine which way the network frames would be transferred between the end systems.

3. Prototype Pitch Control Application integration with SCARLETT hardware modules and applications

Figure 2 shows one of hardware-software structures of the prototype Pitch Control Application developed by RUTRT within the SCARLETT program. The application should demonstrate if it is possible to use two synchronized actuators deflecting an aircraft elevator's surfaces. An external controller computes the position of the elevator on the basis of flight parameters, pilot input, and implemented control procedures. The application communicates with actuator controllers and sensors via an ADFX bus.

The first (P1) partition includes a Flight Control algorithm (FC) block that collects signals from the Pilot, Aircraft and actuator modules and produces the desired pitch angle signals for the controllers. The last module built into the P1 partition is an Error Estimator. It makes it possible to monitor both communication channels and the quality of the control system during the system run-time (compare [16, 17]). The Error Estimator, and FC modules are separate real-time tasks. The second (P2) partition includes the first position controller algorithm (PID1), running as a separate real-time task. Identically, the third (P3) partition includes the second position control algorithm (PID3). The fourth (P4) partition includes the first velocity controller (PID2) joined with actuator (M) whereas the fifth (P5) partition includes a second velocity controller (PID4) module cooperating with a software emulated actuator. The abovementioned partitions are being allocated to appropriate hardware modules by a system integrator. Finally, the software modules (partitions) were being successfully integrated with new IMA2G modules developed within the SCARLETT program.

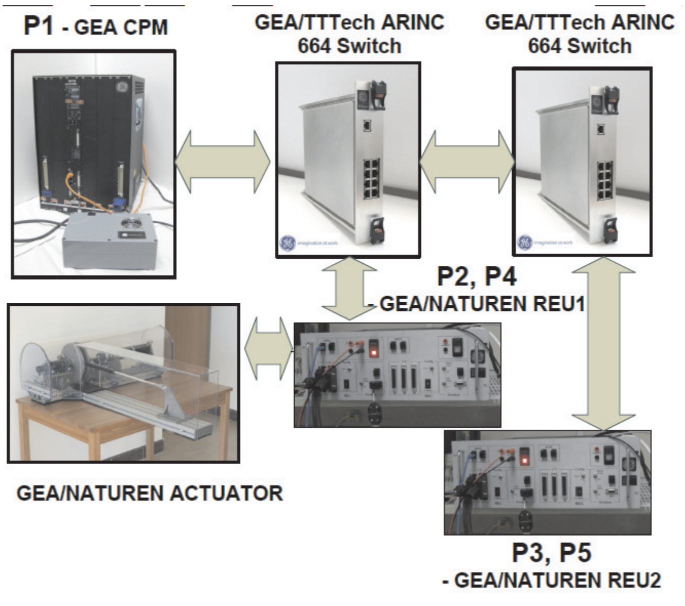

Fig. 3. Hardware modules developed within the SCARLETT program where the prototype Pitch Control Application was integrated with

Figure 3 shows the new hardware modules developed by SCARLETT program partners. The partition identifiers mentioned in the figure indicate the hardware modules which the partitions are integrated with. The P1 partition software module is integrated with the CPM (Core Processing Module) developer by General Electric Aviation – GEA. Apart from pitch control application GEA CPM executes a set of other applications. One of them is the braking system developed by Messier-Bugatti. P2 (P3) and P4 (P5) partitions software is being integrated with REUs (Remote Electronic Unit) modules developed by GEA and NATUREN. ARINC 664-compatible switches developed by TTTech and GEA are responsible for inter-module communication. One of the REU modules is being integrated with a laboratory set including a brushless motor, a load and a power controller developed by NATUREN. The integration of the whole hardware-software system was coordinated by Airbus France.

The integrated system (the demonstrator) was being applied to the evaluation of both hardware and software modules prototypes. Figure 4 includes the result of the pitch control system evaluation. A position tracking by the motor as well as its velocity were examined during experiments. As a result, it was stated that the system met the typical requirements of the position tracking control systems. The system latency was below 40 [ms]. Both the CPM and REU hardware module were able to effectively serve the software applications loaded on them.

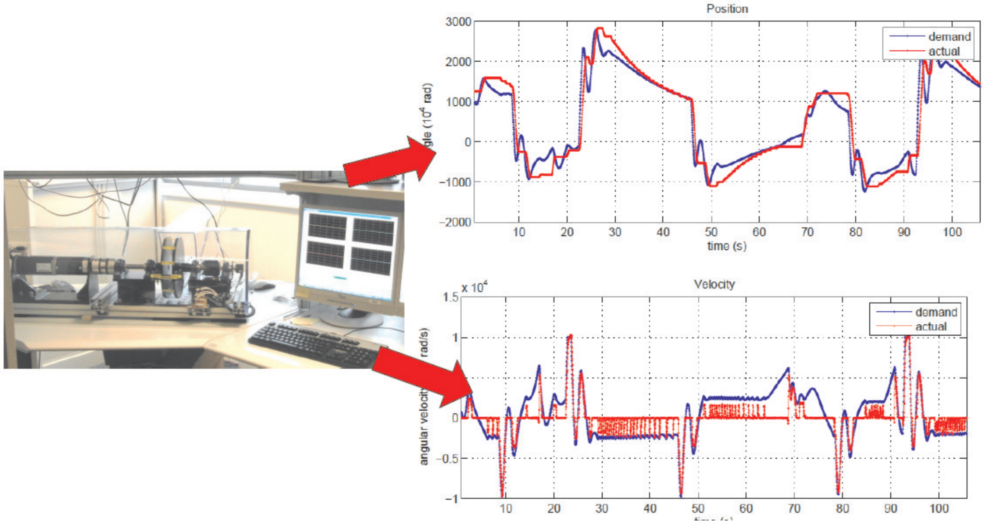

Fig. 4. Position cracking tests of the prototype Pitch Control Application

To effectively integrate the Pitch Control Application with the demonstrator the RU-TRT had to provide a Demonstrator Integrator (DI) with all the necessary (enumerated in section 2) partition and virtual links parameters. The DI collected all the partition and communication requirements and defined the global configuration files for all switches and end systems.

The final report of the SCARLETT program concluded that the hardware modules developed within the program could be potentially applied as a control system applications platform.

4. Conclusions

This paper reports the integration of a prototype Pitch Control Application with hardware and software IMA2G modules developed within the European founded SCARLETT program. Its main aim is to focus attention on the rules of IMA2G systems integration.

During the system development and integration the following new research paths arose. Firstly, some unpredicted delays in data transmission may occur on the hardware/ software layer “between” ARINC specification 653 and ARIN specification 664 based devices. It is hard to calculate precisely, when the data send to an ARINC 653 port becomes ARINC 664 frame. This "gap" should be examined by a separate research. Secondly, it seems that the IMA2G control systems should be developed having in mind simultaneously hardware, control and real-time systems development paradigms. This approach resembles the control system development approach explained in [6, 7]. Thirdly, the knowledge exchange within the SCARLETT program revealed that the aviation industry is still in need of some software tools that could be effectively used for a unified development of ARINC 653/664 based systems.

SCARLETT project and research reported in the paper are founded by 7th European Framework Project, Grant Agreement No. FP7-AAT-2007-RTD-1-211439. The part of hardware components applied in the research published in this paper was founded by European Union Operational Program – Development of Eastern Poland, project no. POPW.01.03.00-18-012/09.

References

[1] *AFDX: The Next Generation Interconnect for Avionics Subsystems*. Avionics Magazine Tech. Report, 2008.
[2] *Aircraft Data Network Part 7 – Avionics Full Duplex Switched Ethernet (AFDX) Network.* ARINC Specification 664P7 2005.
[3] *Avionics Application Software Standard Interface Part 1–2*. ARINC Specification 653p1–2, 2005.
[4] Burns A., Wellings A., *Real-Time Systems and Programming Languages*. Addison Wesley, 2001.
[5] Buttazzo G.C., *Hard Real-Time Computing Systems – Predictable Scheduling Algorithms and Applications*. Kluwer, 2002.
[6] Grega W., *Metody i algorytmy sterowania cyfrowego w układach scentralizowanych i rozproszonych*. AGH, Kraków 2004.
[7] Grega W., *Robust networked control.* [in:] Challenges and paradigms in applied robust control (ed. Andrzej Bartoszewicz), Rijeka: Tech, cop. pp. 349–372, 2011.
[8] http://www.scarlettproject.eu
[9] http://www.sysgo.com
[10] http://www.windriver.com
[11] IEEE 802.3, *IEEE Standard for Information technology, Telecommunications and information exchange between systems, Local and metropolitan area networks, Specific requirements Part 3: Carrier sense multiple access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications*. 2008.
[12] Parkinson P., Kinnan L., *Safety-Critical Software Development for Integrated Modular Avionics*. Wind River White Paper, 2007.
[13] *PikeOS Fundamentals*, Sysgo AG, 2009.
[14] *PikeOS Personality Manual: APEX*, Sysgo AG, 2009.
[15] Ramsey J.W., *Integrated Modular Avionics: Less is More Approaches to IMA will save weight, improve reliability of A380 and B787 avionics*. Avionics Magazine, 2007. http://www.aviationtoday.com/av/categories/commercial/8420.html.
[16] Rogalski T., Samolej S., Tomczyk A., *ARINC 653 Based Time-Critical Application for European SCARLETT Project*. AIAA Guidance, Navigation, and Control Conference, 08–11 August 2011, Portland, Oregon, USA, paper number: AIAA 2011–6684.

[17] Samolej S., Rogalski T., Tomczyk A., *Detekcja stanów awaryjnych w prototypowym systemie sterowania kątem pochylenia samolotu zgodnym z ARINC 653 i ARINC 664*. [in:] Leszek Trybus, Sławomir Samolej (Eds), Projektowanie, analiza i implementacja systemów czasu rzeczywistego, WKŁ 2011, pp. 459–472.
[18] Samolej S., Tomczyk A., Pieniążek J, Kopecki G., Rogalski T., Rolka L., *VxWorks 653 based Pitch Control System Prototype – Second Approach*. [in:] Jan Gruszecki (Ed.), Wybrane zagadnienia sterowania obiektami latającymi, Oficyna Wydawnicza Politechniki Rzeszowskiej, Rzeszów 2011, pp. 125–133.
[19] Samolej S., Tomczyk A., Pieniążek J., Kopecki G., Rogalski T., Rolka T.: *Prototyp systemu sterowania kątem pochylenia samolotu na platformę VxWorks 653*. [in:] Leszek Trybus, Sławomir Samolej (Eds), Metody wytwarzania i zastosowania systemów czasu rzeczywistego, WKŁ 2010, pp. 411– 420.
[20] Samolej S., *ARINC Specification 653 Based Real-Time Software Engineering*. e-Informatica Software Engineering Journal, Vol. 5, Issue 1, 2011, pp. 39–49.
[21] *VxWorks 653 Configuration and Build Guide 2.2*. Wind River Systems, Inc. 2007.
[22] *VxWorks 653 Programmer's Guide 2.2*. Wind River Systems, Inc. 2007.