

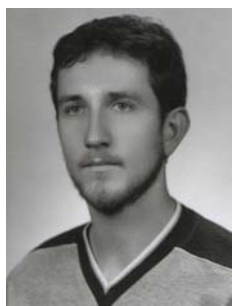
# MONITORING AND DATA ACQUISITION SYSTEM FOR EXPERIMENTAL GENERAL AVIATION FLY-BY-WIRE AIRCRAFT

## P. Rzucidło

Rzeszów University of Technology, Department of Avionics and Control, ul. W. Pola 2, 35-959 Rzeszów, Poland

E-mail: pawelrz@prz.edu.pl

Received 10 12 2004, accepted 07 09 2006



**Paweł RZUCIDŁO**, PhD. Eng.

*Education:* M.Sc. (Eng.) degree in Aeronautics, Rzeszów University of Technology, Faculty of Mechanical Engineering and Aeronautics.

2001 to 2005: Ph.D. studies, Rzeszów University of Technology, Faculty of Mechanical Engineering and Aeronautics, Specialization: Machine Building and Exploitation.

2005 to present: Researcher, Rzeszów University of Technology, Department of Avionics and Control. *Research interest:* Aircraft Control with an Emphasis on Interactions Between Human and Machine.

*Publications:* Author of 4 scientific articles and co-author of 10 conference papers.

**Abstract.** A CAN data bus was used in a project of an experimental Fly-by-Wire control system (SPS-1) mounted on board a PZL-110 “Koliber” general aviation aircraft. This solution allows the free communication between flight control computers and individual modules of the system. The monitoring of the bus state and data frames is useful during particular hardware tests, system integration, laboratory and in-flight tests. This report intends to present the specific monitor tool developed with the to SPS-1. The main part of the CAN monitoring system is software. It realizes acquisition, visualization and conversion of stored data. The monitoring software operates with devices used by an experimental control system and connected to the bus. Particular devices are assigned to separate windows. The operator can control the SPS-1 system through specialized windows or by virtual cockpit. Apart from this, the CAN monitor controls and simulates selected modules of the SPS-1 system and works as the bus diagnostics tool.

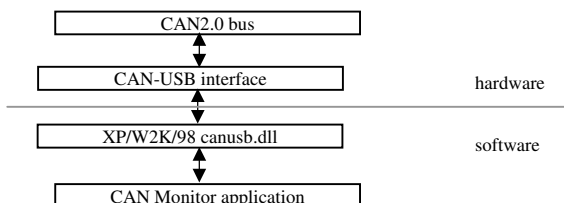
**Keywords:** monitoring, visualization, acquisition, Fly-By-Wire, CAN, in-flight tests.

## Characteristics of the monitoring system

During flight tests the possibility to control the devices being tested is necessary. The monitoring system should be responsible for visualization and data recording and it should allow the free tuning of particular modules, turning on and off of selected devices, and automatic diagnosing of data buses and modules. The solution presented realizes all these functions and allows the free control of the system being tested from the level of the operator of the experiment.

The monitoring system is built on the base of a standard PC, serial communication bus CAN2.0, and CAN-USB interface. The CAN standard was developed in the 1980s for the automotive industry [1]. This standard has since been disseminated and is now used in many fields of technology. It defines the equipment side and data transmission methods and is a way of solving conflicts and detecting bus failures. Units and tools necessary for its service are generally attainable and this standard is characterized by good performance. A CAN bus, version

2.0, was used in an experimental Fly-By-Wire control system (SPS-1) mounted on board a PZL-110 “Koliber” aircraft (Fig 2) [4]. A personal computer is connected to the CAN bus by a standard CAN-USB interface (Fig 1). Requirements for the PC on which the monitoring software works are few and they are restrained only to USB link. Past tests have showed that a microprocessor with a 600MHz clock, 32MB of RAM, and a quick hard drive is fully satisfactory. The monitor worked both on a desktop computer during laboratory stand tests and on a portable one during in-flight tests.



**Fig 1.** Interconnection of the CAN monitoring hardware and software

The monitoring software was made in the Visual Basic environment. The access of the application to the bus is achieved with the help of a suitable dynamic link library installed with the CAN-USB interface [2, 4]. The *CAN Monitor* application can be run directly from its basic directory. The monitoring software contains several subdirectories with help files, additional programs, scripts, and source files. The mutual relations of the hardware and software are presented in figure 1.

The *CAN Monitor* application consists of a main window and secondary ones assigned to it (Fig 3). They are divided in terms of functions (status, filtration, recording) or kind of device-operated device (Tab 1). Calling up the selected window is done by short cut from the keyboard or by menu selection. The main menu is divided into the following detailed menus:

- diagnostics,
- devices,
- simulators,
- recording,
- cockpit.

This version of *CAN Monitor* software is adopted for SPS-1 CAN bus monitoring and data acquisition [6, 7]. The statement of devices operated via on board data buses is presented in table 1.

### The diagnostics

The *CAN Monitor* computer program enables emergency states of data bus and CAN-USB interface to be

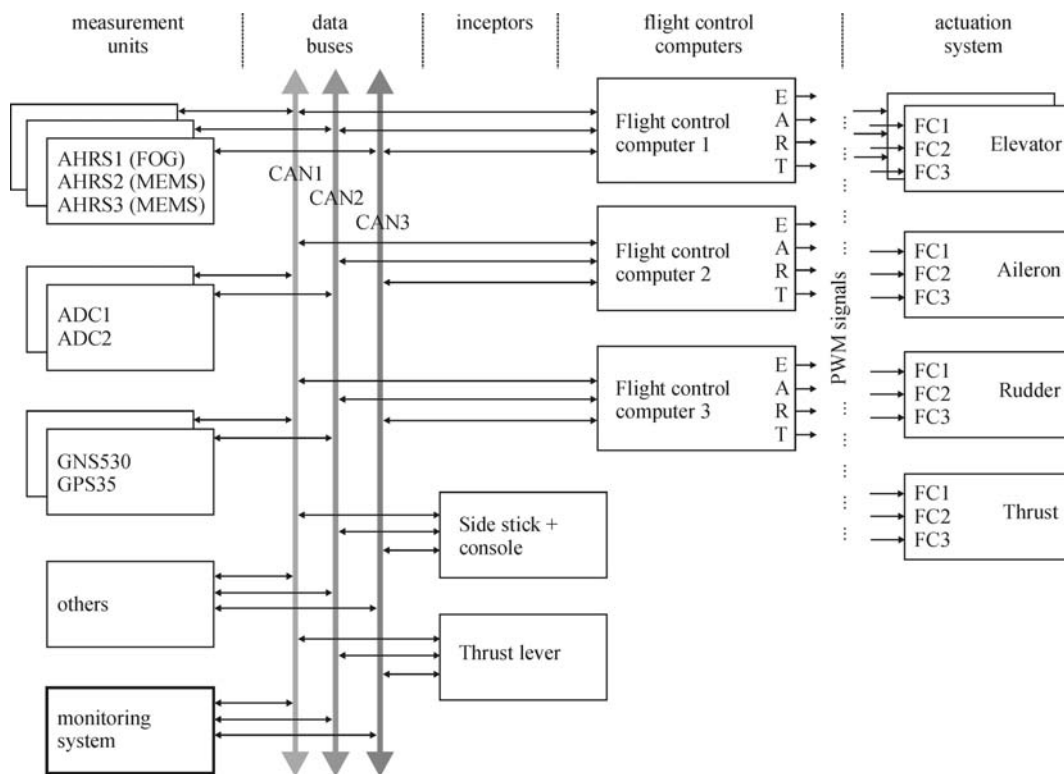
detected. The program informs the user about system failures or the incorrect work of devices in case of:

- data transmission errors,
- unknown data frames,
- isolation of the bus,
- loss of communication between the interface and data bus,
- loss of communication between the interface and PC,
- interface FIFO buffer overflow,
- incorrect installation of the CAN-USB interface.

The bus and the interface statuses are displayed as text messages. A red indicator and question mark next to the frame identifier signal undesirable data frames. Using the filtering function can lock unknown data.

**Table 1.** Operated devices statement

Name	Number	Code	Communication frequency [Hz]
Air data computers	2	ADC	5
Attitude heading reference systems	3	AHRS	20
Control levers / consoles	2	CON	10
Satellite navigation systems	2	GPS GNS	1
Flight computers	3	KS	10
Throttle control	1	ZN	10



**Fig 2.** Structure of SPS-1 system

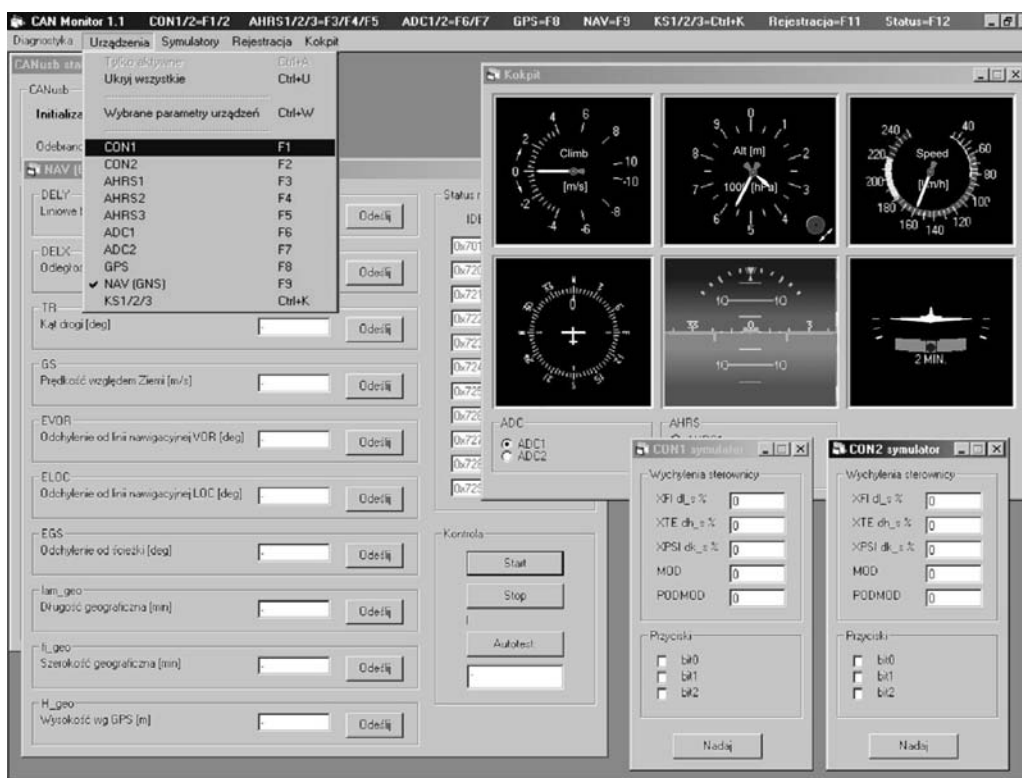


Fig 3. Main window of CAN Monitor application

## Device monitoring

The monitoring software operates with devices used by an experimental control system and connected to the CAN bus (Tab 1). Particular devices are assigned to separate windows (e.g. Fig 4) and the parameters of the SPS-1 modules are displayed there.

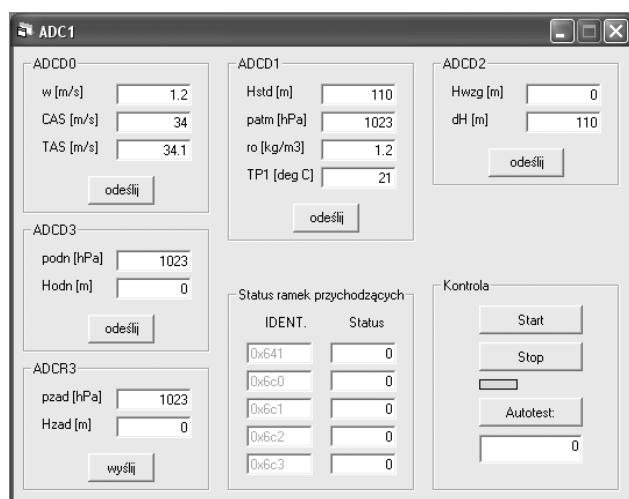


Fig 4. Air data computer (number 1) - monitoring window

Depending on the hardware module work mode, parameters are transmitted continuously or a single data frame is sent on request. *Start/Stop* buttons placed in every window are used for transmission control while an

*Autotest* button allows a self-test of the specific device to be started.

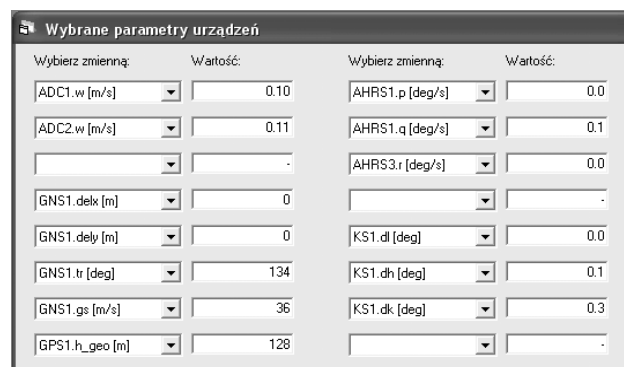


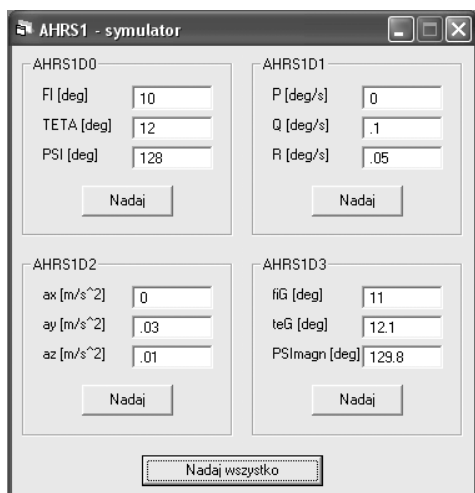
Fig 5. Fragment of window containing chosen equipment parameters

Each window has a graphical marker allowing one to quickly assess whether the device is sending data (movable green bar). Diagnostics fields informing about the relevant measuring frames status also exist.

Traditional monitoring of device parameters during in-flight tests is possible (Fig 4). It is comfortable to monitor two modules at the most. If a greater number of devices are monitored comparing the parameters is rather arduous, demanding many measures and distracting the operators attention. To solve this problem an extra window has been added. This window can contain various parameters from different devices, and the operator can set free configuration of it (Fig 5).

## On-board devices simulation

Building on-board device simulators was necessary during the early stage of SPS-1 research. At that time, a large majority of SPS-1 components were under construction. Simulators allowed early and efficient tests of particular devices and allowed interconnections with non-existent hardware to be checked then. Simulators facilitated the detection of failures and made laboratory stand tests easy [3].



**Fig 6.** Attitude heading reference system (number 1) - simulator window

Simulators of devices are sent data frames identical to data transmitted by real devices. It causes the simulator to be treated as real hardware by other parts of system. The data transmission protocol applied in an early version of the SPS-1 system was characterized by identifiers assigned to defined devices [7]. Each particular device was transmitted frames on the CAN bus with unique identifiers. The version of the SPS-1 system actually developed was modified. It is based on *CAN Aerospace* data transmission protocol, but the principle of how the simulators work is not essentially different [5].

## Data acquisition and recording

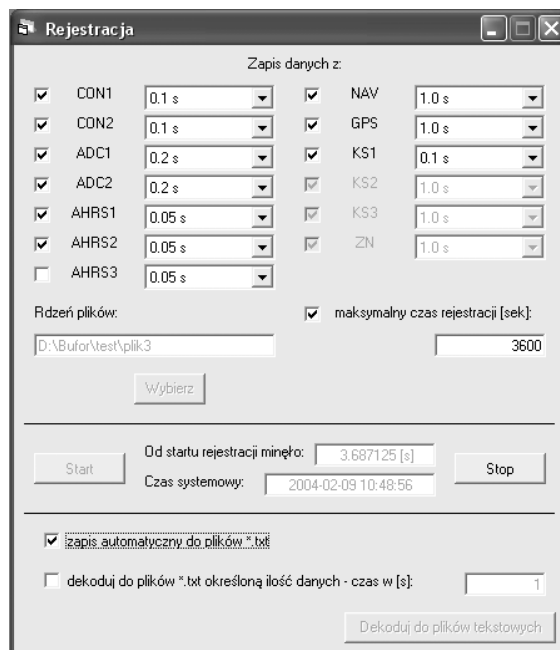
The monitoring software enables data acquisition and recording. Data delivered from all devices forming the SPS-1 system and connected to the CAN data bus can be collected (Tab 1). The operator can select particular devices and frequencies of data recording ( $1 \div 20\text{Hz}$ ) according to needs.

The data received are continuously recorded to files. Each particular file contains data from only one minute of recording and the number of files is equal to the number of minutes. This solution eliminates total data loss caused by unexpected software or hardware failure. All files have an identical name header extended by time of recording (minutes). The name of the header is selected by the name of the operator and the suffix is completed automatically during data acquisition. The software

eliminates the possibility of data overwriting mistakes. The operator can only select a unique file header in the destination directory. Recorded data can be erased from the level of the superior operating system only. *CAN Monitor* enables setting a maximal time of recording. Acquisition is stopped after exceeding this time.

Data received from the CAN bus is saved to binary files. Measurements are written with information about the time-received time. Binary files can be translated directly to text files after recording has been stopped. The text files created are associated with defined devices, for example: con1.txt, con2.txt, ahrs1.txt, etc. The first line of the text file describes the parameters and units recorded. The second line includes the date and exact time recording starts. Data are located in separate columns. The first column contains time counted from the start of recording to the moment data received. The next columns contain values of adequate parameters. An example of text a file decoded from a binary file recorded during an in-flight test is presented below:

```
%t[s], w[m/s], CAS[m/s], TAS[m/s], h[m], p[hPa],
ro[kg/m3]
%03-10-24 12:53:05
.09825 .55 7.5 7.6 193 990.2 1.22
.31925 -.01 7.5 7.5 193.4 990.2 1.22
.52925 .09 7.5 7.6 192.6 990.3 1.22
```



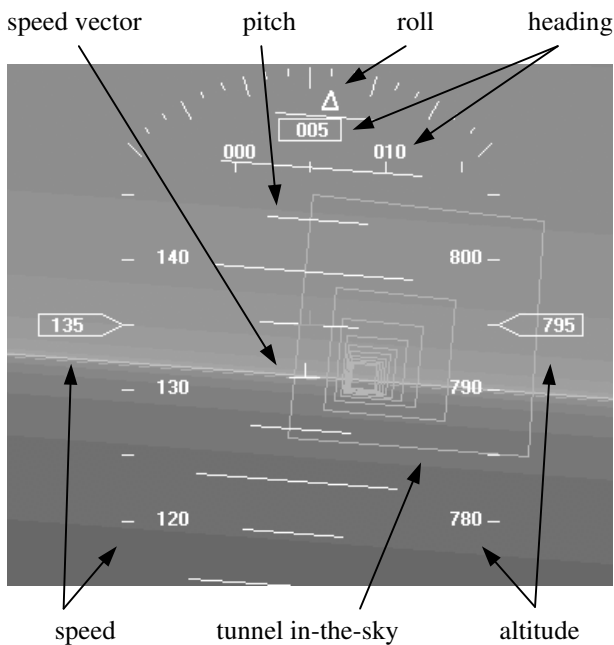
**Fig 7.** Recorder window

The acquisition software requires a minimum of activity from the operator. The recorder can be simply configured and started just before an experiment. During laboratory or in-flight tests, software informs the user about the time of recording.

A complete data set collected during one hour of tests from devices presented in table 1 requires 4.7MB of memory. Measurements can be stored on hard disk drive, USB flash disk, or other fast-access memory.

## Virtual cockpit

An additional function of the SPS-1 CAN bus monitor is virtual cockpit (Fig 3, Fig 8). It enables easy tracking of important flight parameters delivered from a selected air data computer, one of three attitude heading reference systems, and from the throttle control module. Flight parameters like true airspeed, climb rate, barometric altitude, magnetic heading, and attitude are showed on simulated standard aircraft indicators. The operator can also adjust virtual gauges. After every modification the setting of the indicator is sent to the proper device through the CAN bus as data frame. For example, after modification of QNH pressure, the appropriate data frame is sent to the selected air data computer.



**Fig 8.** EFIS indicator as element of virtual cockpit (version actually developed of monitoring system)

## Conclusions

The *CAN Monitor* computer program was utilized for data acquisition and analysis during laboratory stand tests and in-flight tests of a Fly-By-Wire control system mounted on board a PZL-110 “Koliber” aircraft. Twelve hours of experimental flights were recorded to binary files and converted to text files. The total amount of data collected is about 402MB.

Supplements of the *CAN Monitor* are external scripts. They are used for data approximation to equal time periods and for rapid drawing of charts. Connections between scripts and main application are flexible. Graphic user interface (GUI) makes it easy to operate.

The version of the monitoring system presented is not final. This solution is closely connected to general aviation Fly-By-Wire aircraft research and will be developed in the future. Open and modular construction of the

*CAN Monitor* make its further modification and extension possible [2].



**Fig 9.** PZL-110 “Koliber” aircraft preparing for experimental flight

In order to achieve a more reliable solution and perform more advanced real-time tests, two additional CAN type buses will be used. These buses will be applied for communication between flight computers and electro-mechanical servos of control surfaces. The next task will be designing special software for calibration of servos and testing the properties of the actuation system operating on this concept.

The version of the SPS-1 system actually developed is based on *CAN Aerospace* data transmission protocol [5].

## References

1. CAN Specification Version 2.0, Robert Bosch GmbH, 1991.
2. **Pieniażek J., Rzucidło P.** Mikrokomputerowe systemy pomiarowe dla potrzeb lotnictwa, Konferencja Awioniki (Microcomputer Systems for Measurement in Aviation), Waplewo 2001 (in Polish).
3. **Rogalski T., Dołęga B.** The Laboratory Stand Intends to Test and Prototype Control System for Small Transportation Aircraft, International Multidisciplinary Conference, Baia Mare 2003.
4. **Rzucidło P.** Monitor magistrali CAN systemu pośredniego sterowania samolotem PZL-110 (Monitor of CAN Bus Applied in the Fly-By-Wire Control System of PZL-110 Aircraft), *Diagnostyka PTDT 2(32) / 2004* (in Polish).
5. **Stock M.** CAN Aerospace Interface specification for airborne CAN applications V 1.6, Revision 1.6, 12.01.2004, [www.stockflightsystems.com](http://www.stockflightsystems.com).
6. **Tomczyk A.** Facilitated Airplane – project and preliminary in-flight experiments, *Aerospace Science and Technology* 8(2004), [www.elsevier.com/locate/aesctc](http://www.elsevier.com/locate/aesctc).
7. Zintegrowany systemy pośredniego sterowania lekkim samolotem dyspozycyjnym (Integrated Control System for General Aviation Aircraft). Polish State Committee for Scientific Research 8 T12C 049 20 / Ed. A. Tomczyk. – Rzeszów, 2003 (in Polish).