Low-Cost Avionics System for Ultra-Light Aircrafts

Ondrej Spinka, Jan Krakora, Michal Sojka, Zdenek Hanzalek
Department of Control Engineering, Faculty of Electrical Engineering
Czech Technical University
Karlovo namesti 13, 121 35 Prague 2, Czech Republic
spinkao@fel.cvut.cz, krakorj@fel.cvut.cz, sojkam1@fel.cvut.cz,
hanzalek@fel.cvut.cz

***Abstract*— A low-cost, integrated avionics system for ultra-light airplanes is being presented in this paper. It shall represent affordable, yet modern and reliable alternative to the obsolete avionics currently widely used in the ultra-light aviation. Whole avionics is designed as distributed, hierarchical data acquisition system, utilizing Linux operating system and Controller Area Network industrial bus. Important data are provided to the pilot via multifunctional screen, while safety-critical warnings are also reported acoustically via speech system connected to the pilot's headphones. A lot of precautionary measures had been taken to make the system as fault-tolerant and safe as possible. Rugged, robust and reliable apparatus should be the result.**

I. INTRODUCTION

Ultra-light aircrafts have been very popular among sports pilots for decades now, mainly because of their cheapness and unpretentiousness. Unfortunately, they are equally well known for their relative dangerousness, compared to their full-scale counterparts. This is often caused by incompetent operation and maintenance, but many times it is also because of bad construction and poor avionics. In fact, avionics is often the weakest point at all, being relatively costly compared to the rest of the plane. Hence obsolete systems are often being used for financial reasons, naturally having negative influence on the overall operational comfort and safety.

Fig. 1. Development autogyro

The goal of the project presented here was to design affordable, yet robust and reliable modern avionics system for ultra-light aircrafts, incorporating all important functions and features. Because of benevolence of the rules governing ultra-light aircraft design and operation, the certification process of such a system is a great deal simpler and cheaper compared to the certification of avionics for "big airplanes". Therefore it is possible to design relatively sophisticated system while keeping the costs within reasonable limits.

The system was tailored specifically for the new-generation of ultra-light autogyro produced by a local company [1] (see the figures 1 and 2), but can be easily adopted for any type of aircraft. It is a *distributed data acquisition system*, providing pilot with all important flight data such as barometric altitude, true altitude above the terrain (in the range of 0-6 meters), variometer, air speed, rotor RPM, engine data (engine RPM, water and oil temperature, oil pressure, fuel level etc.). All data are displayed on the multifunctional screen and some of them are also provided acoustically to the pilot's headphones (for example the true altitude above ground is reported in form of beeps of variable length, to help the pilot handle the landing approach). It also incorporates plenty of safety functions, such as ground proximity warning, steep descent warning, low fuel/oil/water and fuel/oil/water leak warning, exceeding of speed/rotor RPM/engine RPM warning, engine temperature warning etc. All warnings are displayed on the pilot's screen and also reported by voice into pilot's headphones. The avionics is also equipped with never-ceasing *self-diagnostic process*, which would report any faulty sensor/component within the system. All errors and warnings are logged in the non-volatile memory for further processing. The system also allows to record the course of flight, which is very useful for post-flight briefing. Other data, important for service purposes, are recorded in the long-term. That is especially the engine data, such as maximal revs, water/oil pressures and temperatures etc. Those data are provided to the service technician, and partly also provided to the pilot on-line. For example should the engine require routine service, or the oil needs to be changed, the warning will be displayed on the multi-purpose screen and after the landing, the engine would not start until the maintenance is carried out and the system is re-set by authorized personnel.

The avionics is designed as distributed, hierarchical data acquisition system, with independent intelligent sensors distributed around the airplane, connected to the serial bus, and a central node, called the "main computer", providing essential services. The architecture is described more thoroughly in the rest of this article.

Fig. 2. Cocpit view

The paper is organized as follows: In the next section current modern avionics systems for ultra-light aviation are discussed. Then the exact system specifications are given. In the section IV the hardware architecture of the system is being presented. All components (sensors, embedded computer, data acquisition modules etc.) are briefly described. Then, in the section V, the software scheme is shown and explained. Operating system and HMI (Human-Machine Interface) is also explained here. The section VI describes the system testing. In the last section the conclusions and future work are noted.

II. RELATED WORK

There is relatively wide range of avionics manufacturers specialized on the ultra-light aviation field. Let us name at least some of them. *Dynon Avionics* (http://www.dynonavionics.com), a well-known US company, produces systems similar to the one presented here. They provide probably the most sophisticated avionics for the ultra-light field overall. *Braeuniger Flugelectronic* (http://www.braeuniger-flugelectronic.de) is a German manufacturer, producing cheaper and simpler devices compared to Dynon. *Flytec* (http://www.flytec.com) is a Swiss company, whose products are targeting mainly the hang-gliding and paragliding market. A good overview of other manufacturers is given by the *Ultralight Avionics Server* (http://www.ulavionics.cz). According our knowledge, no manufacturer currently produces complex, sophisticated avionics solutions comparable to the one presented in this paper. Even the Dynon Avionics, although providing complex avionics solutions, cannot match this system with their current products, as they provide more or less stand-alone devices, they do not incorporate data logging and audio interface etc.

III. SYSTEM SPECIFICATIONS

The system specifications were briefly described in the section I, but let us now summarize its features in a more systematic manner. Following functions were planned (some of them are not implemented yet):

- Basic flight data (air speed, barometric altitude, variometer, engine and rotor RPM, fuel level, oil and water temperature, oil pressure)
- Audio warning system - safety-critical issues are reported acoustically to the pilot by spoken language: water/oil/fuel leak, water/oil/fuel low level, excessive engine/rotor RPM warning, critical rate of descent
- Acoustic proximity warning system - true height over terrain, determined from the ultrasound sensor (in the range from 0 to 6 meters), is signaled to the pilot by a series of beeps of variable length (similar to car parking proximity warning system)
- Data logging system
- Self-diagnosis system, faulty sensor detection
- GPS navigation

On top of that, the system should be easily-extendible, robust and reliable, and financially affordable. From the features mentioned above almost all have been implemented, safe for the GPS navigation and some of the data logging functions. Also the self-diagnosis functions can be further extended.

IV. HARDWARE ARCHITECTURE

A. *Overall system design*

The main idea was to design a system that would be easy to extend and that would require as little wiring as possible. The logical decision was to build a *distributed system* [2], consisting of relatively autonomous nodes, interconnected via industrial serial bus. This architecture is widely used nowadays, being convenient from many points of view. Its main advantage is that it significantly simplifies the wiring compared to traditional architectures, therefore reducing the weight and installation costs. Another asset is that the functional blocks (nodes) of the system can be designed and tested separately, which considerably simplifies the design/testing process. The system functionality can also be extended easily, as other nodes may be included and connected to the bus with relatively little fuss. And last but not least, since the nodes are independent, the failure of one of them would not necessarily compromise the functionality of others and would not cause the whole system to fail. This feature significantly contributes to the overall system reliability.

The block scheme of the system is shown in the figure 3. The *main computer* is the core member, providing the crucial functionality - that is the data visualization and storage, and the human-machine interface. The other members serve as data acquisition elements and operate independently. As various sensors with different interfaces were used, the *data acquisition nodes* (see the section IV-D) were necessary to provide a standard interface. Their purpose is to acquire data from the sensors using various means, and to send those data via the bus to the main computer. Each of these modules is placed as close to the sensors it operates as possible.

The *Controller Area Network* (CAN) (see [3]) is being used as the interconnection medium between the nodes. It was chosen because of its well-known reliability and fault tolerance. The CAN is a proven industrial standard, widely used mainly in the automotive industry. It is a differential

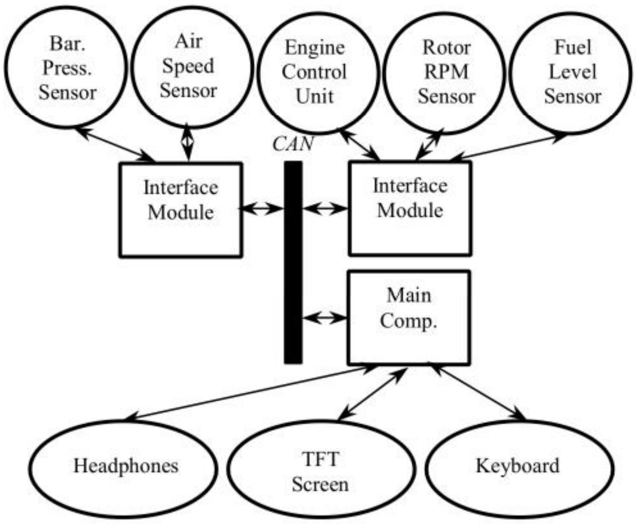

Fig. 3. Overall block scheme of the system

serial bus, noise-resistant and fault-tolerant. Another important feature is the strictly deterministic bus assignment policy, with non-destructive packet collision solving. That is especially important for real-time communication (see [4]).

B. Sensors

The correct sensor selection was probably one of the main concerns of the whole project. It was necessary to choose small and lightweight, yet reliable and robust industrial sensors, preferably certified for avionic use and equipped with digital output. Unfortunately, it turned out that to find sensors satisfying all those demands within reasonable price range is almost impossible. To fulfill at least the most important requirements, certified lightweight robust sensors were chosen, unfortunately utilizing different types of outputs (voltage current, duty cycle and digital (RS-232 and RS-485). Therefore a need arose to design a *Universal Data Acquisition Module*, which would be able to handle all those different types of signals, to preprocess the data and to send them via CAN bus to the main computer (see the section IV-D for more detail). Such modules could be placed near the respective sensors, to provide standardized output services.

Let us now briefly describe the sensors that were finally chosen. *Honeywell ppt0015AW5VB* absolute pressure sensor (see the figure 4) [5] is used to measure barometric pressure, which was later used for barometric altitude and altitude difference (variometer) computation. It is an intelligent, internally calibrated sensor with internal temperature compensation, featuring RS-485 output.

Honeywell 946-A4V-2D-2C0-65E ultrasonic distance sensor [6] serves to measure precise true altitude above the ground during landing and to generate ground proximity warning. The sensing range is 0.5 to 6m. The sensor is mounted to the airplane frame such that it is positioned 0.5m above the ground when the aircraft stands on the landing gear, to avoid the "dead zone" of the sensor.

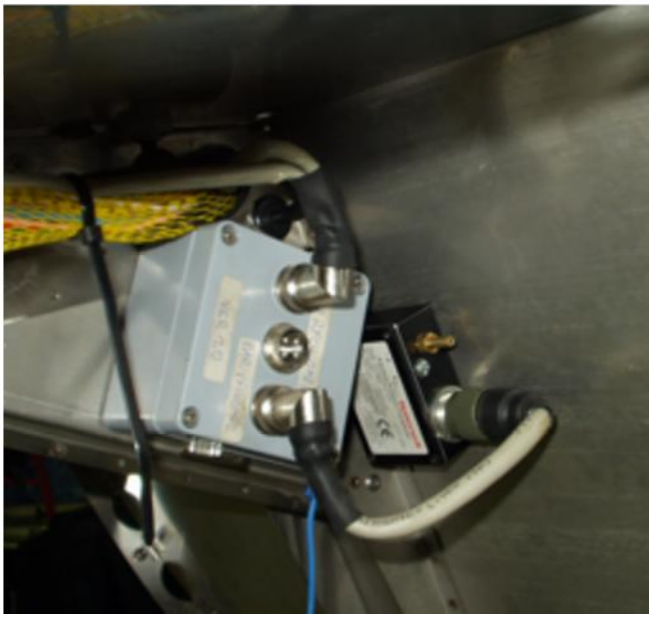

Fig. 4. Honeywell ppt0015AW5VB barometric pressure sensor with interface module

Fig. 5. Honeywell 3030 VRS RPM sensor measuring the main rotor RPM

To measure the rotor RPM the *Honeywell 3030 VRS* (Variable Reluctance Sensor) (see the figure 5) [7] is being used. It is a simple, rugged self-powered device, positioned opposite to the steel tooth gear on the main shaft, which generates a pulse whenever a tooth comes by.

The *Flytec paragliding airspeed sensor* [8] is used to measure the airspeed of the vehicle. It is a propeller-in-the-pipe type of sensor, where the RPM of a small propeller driven by the airflow is being measured.

The engine-data (engine RPM, oil temperature and pressure, water and air temperature, fuel level and main battery voltage) are provided by the *Ignitech IgniJet ECU* (Electronic Control Unit) [9], which was specifically developed for the ultra-light aircraft two-stroke engine control.

C. Main computer

The *MSM-586SEV* [10] industrial embedded PC-104 from *Digital-Logic* is used as the main computer. It is an embedded PC, based on the *Elan 520* microprocessor running on 133MHz. Its most important features are 64MB SDRAM, CompactFlash socket and VGA output with digital display controller. Additional CAN bus controller and Sound card from Digital-Logic is used to extend its functionality. The computer is mounted in an industrial, robust protective housing. The housing is attached to the frame using rubber vibration-damping silent-blocks (see the figure 6).

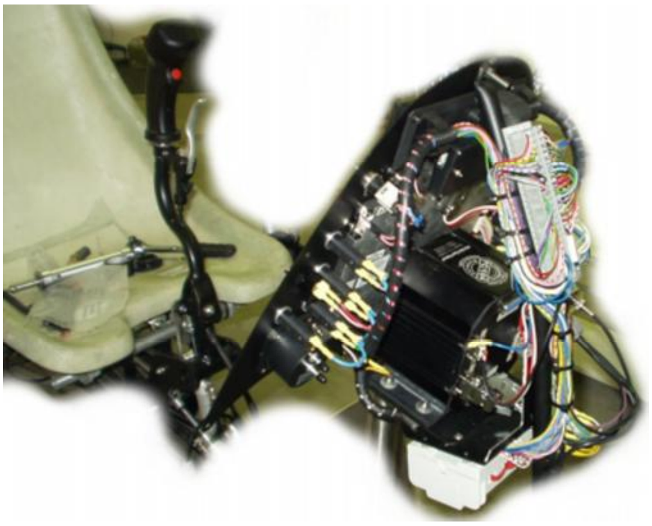

Fig. 6. PC104 in protective housing and facial panel wiring

The *NEC NL6448BC33-59* TFT color LCD display [11] with *NEC 104pw161* backlight inverter is used as the visualization device. It is equipped with anti-reflex stripe to prevent reflections.

D. Sensor interface module

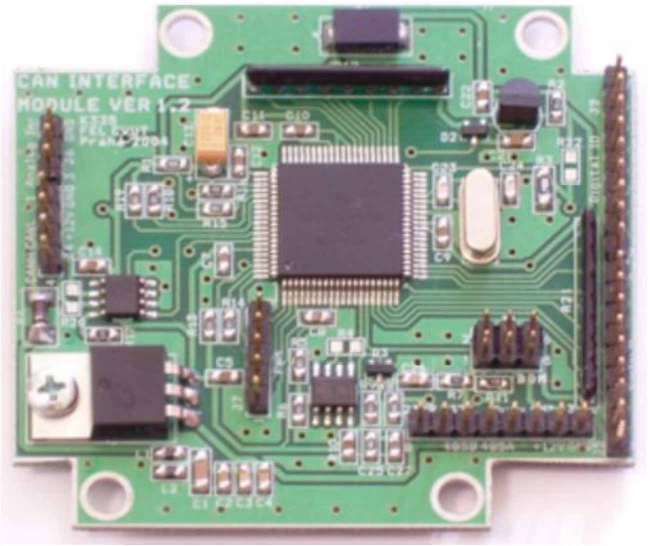

Fig. 7. Sensor interface module

As the avionics incorporates sensors with different types of output (voltage or current, duty cycle, RS-232 or RS-485), a need arose to design a *Universal Interface Module*, which would be able to handle all those different types of data. Main purpose of this module is to periodically acquire new data from the sensors, to preprocess them and to send them via CAN to the main computer in a unified format. It is simple and cheap, robust and reliable and universal in terms of wide range of input signals it can handle.

The module (see the figure 7) is built around the *Motorola HC12* family microcontroller, namely *MC68HC912D60A* [12], featuring 60kB of FLASH, 2kB of RAM and 1kB of EEPROM. It features 16 digital inputs/outputs, 4 PWM outputs (could be also configured as additional digital IO), 4 analogue (voltage or current) inputs (can also be configured as additional digital IO), 1 duty cycle/PWM input that could handle DC signals with amplitude from 0,15V to 30V and arbitrary DC offset, independent CAN, RS-232 and RS-485 lines (all of them could be operated simultaneously). It can be powered by un-stabilized 6,5V-30V DC voltage. The module provides stabilized and filtered noise-free 5V and 12V power source to the sensors.

For safety reasons, the module is equipped with *time and voltage watchdog*, which would reset the processor should the software hang-up occur or should the input voltage drop below the required level. The digital IOs are equipped with pull-up resistors and protective diodes, to prevent its damage by applying excessive voltage.

E. System safety

Safety of the whole system is one of the most important issues that have to be solved when designing avionics, even for small ultra-light airplanes. Let us now briefly describe precautionary measures we have taken to maintain fault-tolerance of the system. As was described above, failure of neither node (excluding the main computer) might cause overall system collapse. Instead of that, each failure is immediately reported on the display and logged to the flash memory. Should the faulty node recover again, its function is restored immediately, but the error is logged anyway. After each flight, this log should be checked to find out whether anything went wrong with the system during the flight.

Each sensor is checked periodically by respective data acquisition node. The data provided by the sensor must fall within pre-set boundaries and also the difference between adjacent values must be less then pre-set threshold. Should any of those conditions be violated or should the sensor stop responding to the data acquisition node, it is reported as faulty.

Every system node (data acquisition nodes, ECU and main computer) is equipped with *time and voltage watchdog* to prevent software hang-ups. Any hang-up that would occur is detected and logged into the error log file.

The whole hardware, either provided by external suppliers or built in-house, does of course satisfy industry norms on temperature operating ranges and electromagnetic compatibility. Robust industrial wirings, connectors and housing were used to achieve desired robustness and reliability of the system.

F. *Electromagnetic interferences related issues*

Unwanted electromagnetic interferences are one of the most common and hard-to-solve issues that one have to cope with when designing avionics. This is caused by various reasons. One of the most important aspects the designer has to bear in mind is the absence of the ground potential on board of an airplane. Often the vehicle frame is being treated as the ”ground”, but this is sometimes a very misleading analogy. Electric potential of the airframe can vary in ranges of tens of volts per meter. This is caused by the friction of the airframe in the air, electric potential of the air and by the operation of the powerful on-board wireless transmitter, which often uses the airframe as the antenna counterpoise. Those effects may lead to serious electromagnetic interactions in the on-board electronic systems, which are not so easy to fix.

In our case, the interactions described above led to the most serious difficulties that we had to solve during the whole design process. We have encountered minor problems with the display when the wireless transmitter was operated, but those issues were solved swiftly by shortening the data cable between the display and the main computer and by attaching ferrite rings to both ends.

A lot more serious interactions arose between the barometric pressure sensor and the on-board wireless transmitter. The sensor was drifting a lot whenever the transmitter was operated. We were a bit surprised by this issue, for we used an airplane - certified, rugged industrial intelligent sensor (see the section IV-B for details) and therefore did not except such a problem to appear. The reason of this behavior was relatively hard to track, for it was caused by two unrelated issues - there was a faulty connector (although it appeared to be fine), and the sensor turned out to be considerably sensible to the noise in the power lines. The problem was solved by adding a filter into the power supply - a self-induction coil and a set of capacitors tuned to dissipate the high-frequency noise did the trick. Ferrite rings were also attached to the cable connecting the sensor with the data acquisition module.

G. *Vibrations related issues*

The ever-present vibrations of the airframe can also lead to the system malfunction or can cause the strange-looking behavior of some system parts. We did not encounter any major vibration-related issues, as we were well prepared for that. For external connections of the system modules (i.e. power lines, bus lines etc.), only robust industrial connectors, fastened by sleeve nuts or bayonet joints, were used. Inside the housings lighter plastic connectors were used, but all were equipped with self-locks and were sealed with silicon glue. Every PCB was coated with a PCB varnish, which has two functions - it fixes the parts to the board, preventing them from dismantling due to high vibrations, and also works as a short-circuit protection (that might be caused by small metal cuts, for example).

The only vibration related issue we had to solve arose in the main rotor RPM measurement. A VRS (Variable Reluctance Sensor) attached opposite to a steel tooth gear is used (see the picture 5). During the engine tests we noticed that sometimes the sensor reports a ”fake” RPM (it was measuring 250 RPM, although the rotor was not moving at all). Interestingly, it always measured exactly the same value (250 RPM). We discovered that at certain engine speed the sensor fixture got resonance, and when a tooth was situated right opposite to the sensor head by chance, it was enough to induce a signal with sufficient amplitude to be registered. This problem was solved by slight mechanical adjustments in the sensor fixture.

V. SOFTWARE ARCHITECTURE

A. *Overall design*

Let us now briefly describe the software concept. From the platform point of view, we may logically distinguish the software to the part running on the main computer and the software of the data acquisition modules. The programs of respective data modules differ according to their purpose (i.e. sensors they operate). The modules are running systemless (using no operating system) and their software is relatively simple. Its purpose is to acquire the data from the sensors in precise time moments (the sampling frequency is generated by the modules themselves, because of the real-time issues that would come into play should the main computer, utilizing no real-time OS, be involved). On top of that, the modules also pre-process the data (computation of barometrical altitude and vario value using the measured barometrical pressure, for example, is carried out directly within the module operating the pressure sensor, to ease the computational burden that would lie on the main computer otherwise). The modules also carry out periodical diagnosis of the sensors they operate.

The main computer is utilizing the *Gentoo* distribution of the *Linux 2.4.20* OS. As the PC-104 platform with Intel 80486 compatible processor is being used, porting of the OS presented no problem. The *LinCAN* driver [13] was used to operate the CAN interface card.

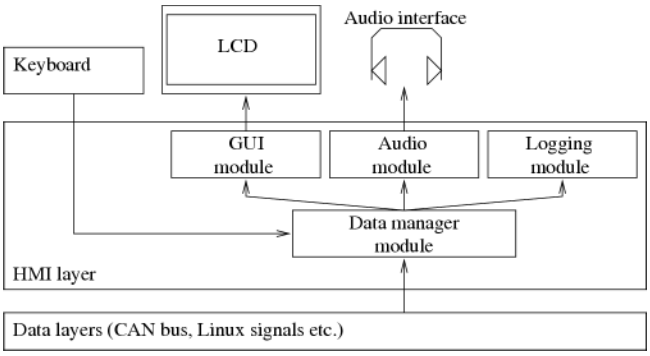

Fig. 8. Software architecture scheme

The software of the main computer consists of two basic layers (implemented as processes communicating via *pipes*) (see the figure 8). The basic is the *Data Layer*, reading data packets from the CAN and passing them along to the *HMI (Human Machine Interface) Layer* in the desired format. The second layer, providing the HMI services, is described in more detail in the next subsection.

B. Human-machine interface

In this subsection the *HMI (Human Machine Interface)* shall be presented (see the figure 9). It is an essential part of the system, which provides the gathered data to the pilot, either visually or acoustically. Let us now briefly discuss its structure (see the figure 8).

Fig. 9. Multifunctional TFT screen

The HMI is a *multi-process application* fully implemented using *GNU ANS-II C*. The internal process communication between lower layer and HMI and among HMI modules is implemented using *pipes* (FIFO). The application consists of four modules. The *Data Manager Module* is responsible for data flow in the HMI. The data are provided either by the data layer or the keyboard. The Data Manager Module retransmits those data into *Graphical, Audio or Logging Module*. The data exchange is provided by *internal messages*.

The *Data Manager Module* is being used to preprocess and redirect data received from lower layers to other modules. The redirection is given by a customer specification. For example all data dealing with the external temperature are transmitted into the GUI Module to be displayed on the screen. All data dealing with the altitude are transmitted to both GUI and Audio modules, in order to be displayed in the graphical form and to provide the audio proximity warning.

The *GUI Module* displays the HMI graphical objects on the screen. Respective object properties are changed in accordance with the received data. The objects deal with azimuth pointer, time or temperature measurement for example. The *SVGAlib* and *Tifflib* libraries were used to handle graphical objects.

The *Audio Module* provides audio services. Similarly to the graphical interface, the data are provided by the data manager module. Sound files are played according to the message data content. The sound volume can be adjusted anytime. The GNU *OpenAL* library has been used to deal with the sound files, for it is easy to configure and it supports a lot of audio formats.

The *Logging Module* stores specific data to a *log file* located in the non-volatile memory. The log file content can be used for post-flight analysis or service purposes. The log is protected by password and can be accessed by authorized personnel only.

VI. SYSTEM TESTING

Before it was mounted to the development autogyro, every system component was tested in a climatic chamber, where temperatures ranging from -35°C to 70°C were reached. As we used only components fulfilling the industrial temperature range (-40°C to 85°C), no problems were encountered, although the system was running for two days non-stop in such a harsh environment. Also ”cold” and ”hot” starts were tested (the system was switched off, left to cool down or warm up to the ambient temperature, and then switched on again).

Unfortunately, we do not posses a vibration test bed, and therefore we were unable to emulate the operational vibrations. During the test flights no vibration related issues were observed.

The overall system functionality was tested only very briefly so far. During the test flights, a reference set of instruments, independent on our system, was carried and the measured data were compared by the pilot (unfortunately, it was not possible to record the reference data, as only analog pointer-type instruments were used). Until now, no other tests were performed.

VII. CONCLUSIONS AND FUTURE WORK

Although the system described here had successfully passed the first flight tests, a lot of work still needs to be done. Most important is more thorough testing, to ensure maximal stability and reliability and to trace and fix possible design flaws. Then some minor adjustments of the Human-Machine Interface according the pilot’s demands are necessary. Afterwards, the system can be extended of additional functionality - first of all, integration of GPS navigation would be useful. A touch-screen display can be used instead of keyboard as input device, although it is not clear whether this would be a good idea from the robustness point of view.

Overall, the system still cannot be treated as ready for regular usage.

On the positive note, it is clear that the concept had shown early promise and it is definitely worth of future exploration. The flight tests did not reveal any major difficulties. There were serious problems with unwanted electromagnetic interactions between the barometric pressure sensor and on-board wireless transmitter, but they were solved successfully. Another minor problems arose due to the mechanical vibrations generated by the engine, yet a solution was found swiftly.

Modern, integrated complex avionics solution such as this clearly presents a major step forward in terms of operational comfort and safety, both for the user/pilot and the manufacturer/service personnel. From the user point of view, the significant benefits are that important flight data are provided in a clear and lucid way thru the display or audio output, important safety-critical warnings are reported in both audio and visual way, the pre-flight routine is being checked by the system and no important steps can therefore be omitted, either deliberately or not. All flight data are recorded and therefore allows to carry out a thorough post-flight briefing, which is especially important for less-experienced pilots. Thanks to this, possible piloting errors may be revealed more easily, helping with the pilot training process and to hone piloting skills.

From the manufacturer/service personnel point of view, it is important that the system records important operational data, such as engine run time, engine temperatures and revs, water and oil pressures and temperatures, main rotor revs etc. It is therefore easier to diagnose and track possible defects and reveal their cause. The manufacturer can easily determine whether the vehicle was operated within required terms all the time. On top of that, routine maintenance, such as regular oil change, is monitored by the system and should any of the required services be omitted, the system could even deny to start the engine until the service is done.

We therefore believe that systems such as this do have a future and are worth of further development and exploration.

VIII. ACKNOWLEDGMENTS

This work was supported by the Ministry of Education of the Czech Republic under Project 1M6840770004.

REFERENCES

[1] ProAir LTD, *SkyWalk GL-165 Ultra-Light Autogyro*, 2005. http://www.proair.cz.

[2] J. Ehrlich, A. Zerrouki, and N. Demassieux, "Distributed architecture for data acquisition: a generic model," in *Proceedings of the IEEE Instrumentation and Measurement Technology Conference*, vol. 2, pp. 1180–1185, 1997.

[3] K. Etschberger, *Controller Area Network*. IXXAT Press, 2001.

[4] L. Pinho and F. Vasques, "Reliable real-time communication in CAN networks," *IEEE Transactions on Computers*, vol. 52, no. 12, pp. 1594–1607, 2003.

[5] Honeywell corporation, *Precision Pressure Transducer User's Manual*, 2000. http://www.honeywell.com.

[6] Honeywell corporation, *946-A4V-2D-2C0-65E Datasheet*, 2001. http://www.honeywell.com.

[7] Honeywell corporation, *3030 Variable Reluctance Sensor Datasheet*, 1999. http://www.honeywell.com.

[8] Flytec corporation, *Paragliding Airspeed Sensor*, 2004. http://www.flytec.ch.

[9] Ignitech corporation, *IgniJet Max Power Datasheet*, 2004. http://www.ignitech.cz.

[10] Digital-Logic corporation, *MSM586SEV Embedded Computer Datasheet*, 2004. http://www.digitallogic.ch.

[11] NEC corporation, *NL6448BC33-59 TFT Display Datasheet*, 2004. http://www.nec.com.

[12] Motorola corporation, *MC68HC912D60A Microcontroller Datasheet*, 2001. http://www.motorola.com.

[13] P. Pisa and F. Vacek, "Open Source Components for the CAN Bus," in *In Proceedings of the 5th Real-Time Linux Workshop, Universidad Politcnica de Valencia*, pp. 215–221, 2003.