Experimental Payload for the PilsenCube Picosatellite

Karel Dudacek

Petr Mayr

Department of Computer Science and Engineering
Faculty of Applied Sciences
University of West Bohemia
Pilsen, Czech Republic
dudacek@kiv.zcu.cz
pmayr@students.zcu.cz

***Abstract* – This paper presents hardware and software design of the payload for the PilsenCube picosatellite. The goal of the payload is to test the suitability of microcontroller with FRAM based code memory in the low Earth orbit environmental conditions. The proposed set of tests comprises FRAM memory testing as well as testing of some selected functional units of the microcontroller.**

Keywords – Picosatellite; FRAM memory; MSP430 MCU; Low Earth Orbit

I. INTRODUCTION

The picosatellite technology brings possibility to send to Low Earth Orbit (LEO) a number of various small experimental devices. Most of these experimental devices use microcontroller to control the experiment itself as well as the data link providing data channel to the ground station and uplink command channel to the picosatellite experiment. These microcontrollers have to work in LEO harsh environmental conditions. They have to withstand some radiation dose and have to be able to work in a wide range of temperatures.

For use in a radiation environment a special variant of electronic devices are available, but its use is often out of limits for (financial) resources available for picosatellite design. However, some studies show that for short time space missions the use of common i.e. not radiation hardened devices is possible [1], [5].

The goal of our picosatellite payload is to study the behavior of selected type of microcontroller in environmental conditions which can be considered to be typical for a picosatellite board.

II. SELECTION OF THE MICROCONTROLLER

A wide range of microcontrollers (MCU) is available for purpose of controlling some physical or technological experiments on picosatellite board. Most of these microcontrollers use Flash memory to store program code. Radiation sensitivity of this type of memory can easily become one of the most serious sources of problems.

Another alternative to Flash memory MCU are microcontrollers with FRAM data and code memory. FRAM memory has shown some radiation tolerance in testing in the space environment [2]. In Texas Instruments MSP430 MCU family we can find some microcontrollers with FRAM code memory.

MSP430 are 16bit RISC microcontrollers, produced with a wide range of various peripheral modules. Advanced low power modes make this MCU sufficient for applications with limited power budget. The computing performance of high end types of this family is high enough for various applications in small satellites, e.g. controlling experiments, communication control etc. [7]. Most of the types use Flash code memory, but a number of types with FRAM memory are also available. Thanks to the above mentioned properties MSP430 family members appear to be suitable MCU for a number of applications onboard small satellites. Having possibility to add small payload to the PilsenCube picosatellite, we decided to examine the behavior of FRAM based types under conditions on LEO.

III. THE EXPERIMENT OUTLINE

The best way to test digital electronic devices would be to use Boundary Scan Technology. Although MSP430 MCU has JTAG port, the Boundary Scan Chain (BSC) is not implemented in these devices. JTAG port is used only for memory programming and debugging purposes. Due to this fact we are limited only to functional testing of the device.

The basic idea of the experiment is as follows: The MCU will periodically execute a set of self-test procedures, testing code and data memory and a set of selected peripheral modules (serial interfaces, AD converters, timers etc.) As one of important features of MSP430 MCU is a set of low power modes, the test will include examination of these modes [6].

Self-testing of microcontroller is based on the assumption that at least the CPU core and a part of memory containing the test program code is working properly. Failure of the testing program itself will be fatal for the whole testing experiment. The failure can have one of main reasons:

- Hard error. In this case part of the CPU or memory is permanently damaged. This error is lethal for the MCU.
- Single Event Upset (SEU). As this is a recoverable error, there is the possibility to

ISBN 978-80-261-0722-4,

restart the test procedure and recover normal device function. Restart can be triggered by the internal MCU watchdog or by the second microcontroller as the response to missing ALIVE_*x* signal.

- Soft error, causing change of some bits in memory, while memory itself is not damaged. While this error cannot be recovered by restarting the test procedure, there is the possibility to rewrite content of code memory by an external programming device and restart the MCU.

As the memory error is supposed to be one of the most probable failure causes, the experimental payload has two identical microcontrollers. Should one of them fail due to memory failure, the second one will try to reboot its memory and restart the program. If both microcontrollers suffer soft error in the same time, the error cannot be recovered and the payload mission will be at the end.

IV. The Experimental Payload Hardware

As mentioned above, the cores of the experimental payload are two identical microcontrollers (see Fig. 2). The Texas Instruments MSP430FR5994 FRAM microcontrollers were chosen. The MSP430FR5994 has 256 kB of FRAM memory, which can be used as code as well as the data memory. The FRAM memory is nonvolatile, i.e. it holds its content when the MCU power is removed.

A. External Watchdog and ALIVE_x Signals

As each microcontroller has the possibility to reset and/or reprogram its counterpart and both microcontrollers share the same downlink communication line, special attention was given to protect "healthy" circuits against unintentional affecting by the "ill" microcontroller. Each microcontroller services its own external watchdog device. This device with some additional logic provides the ALIVE_X or ALIVE_Y signals. (In next chapters we will use generic ALIVE_*x* notation for these signals.)

B. Reset and Bootloader Pins

If one of microcontrollers detects malfunction of the second one (by inactive input of ALIVE_*x* signal or by cease of periodical status reports from it as will be described later), it has the possibility to try to resuscitate it. Each microcontroller can control reset pin and bootloader pins of its counterpart, so it can try to reset the other one or to reprogram its code memory. Program control of the RESET pin and bootloader start TEST pin is possible only if ALIVE_*x* signal is in active level, thus protecting against its unintentional activation by "ill" microcontroller.

C. Downlink Interface

The payload board has one downlink SPI interface, shared by both microcontrollers. Both ones are configured to be SPI slaves. In any time only one of the microcontrollers is in active state, responding to write and read commands from SPI master [4], [8]. As well as RESET and TEST pins the SPI outputs of both microcontrollers can be in active state only if the respective ALIVE_*x* pin is in active level.

D. Other Components and The Board Layout

Besides the two microcontroller devices the experimental payload board contains small number of additional logic circuits. Though the producer does not state it, these devices of 74LVC family were proofed to be radiation tolerant [5].

The PilsenCube Picosatellite is the cube of size approx. 100 × 100 × 100 mm (see Fig 1). The payload board of the size approx. 25 × 100 mm is placed in one of the inner floors of the satellite PCB stack. This brings some level of protection of the board against extreme temperature changes – it will not suffer direct Sun radiation or thermal radiation to the open Space. We expect the temperature will not exceed the common temperature range of industrial class of the components. This assumption is based on the available VZLUSAT onboard temperature data [9]. (The design of some another PilsenCube electronic modules is based on this assumption as well).

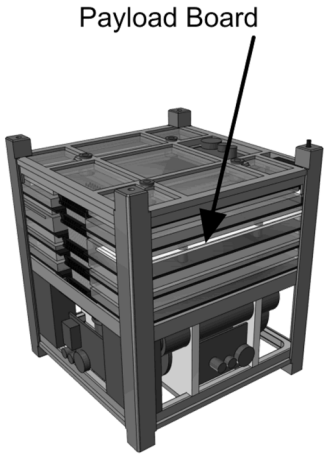

Figure 1. The PilsenCube picosatellite and the experimental payload board placement.

V. The Experimental Payload Software

As mentioned above, the experimental payload board contains two MSP430 microcontrollers. The interconnect line between them makes possible the bidirectional communication.

A. The Software Outline

Both microcontrollers run identical software stored in the internal FRAM memory. One of the microcontrollers is the master while the second is the slave. After reset, the dedicated pin makes possible to detect if the microcontroller is MCU_X or MCU_Y. The default master or default slave role is assigned to them accordingly. Master and slave programs both execute state automaton, which control the program flow according to the state messages and commands received on interconnection line and on timer signals. Generally, the slave executes self-test procedures and sends state information to the master. In the same time master checks slave operations and execute recovery actions if slave operation fails. After one test period both microcontrollers exchange their master and slave roles.

B. The Master and Slave Roles

After reset, MCU_X become default master while MCU_Y become default slave. The role of the master is as follows:

- Master receives state messages from the slave and stores it to log records. Log records are stored in the FRAM memory.
- Master checks timeouts of self-test operations of slave. If slave will not confirm end of execution of some self-test procedure by appropriate message until timeout expires, the master supposes that slave program execution failed. Master then starts some slave's recovery procedures.
- Master has control of the downlink SPI data line. As the payload is SPI slave on this line, the role of the master (in sense of master-slave microcontrollers' roles) is to response to the SPI master read commands.

The role of the slave is as follows:

- The slave executes step by step self-test procedures. At start of each procedure it sends a message to inform the master. At the end of the procedure the slave sends a message containing the results of the just finished self-test procedure.
- Slave periodically sends I_AM_SLAVE messages to inform the master that it is alive and in slave mode. It checks I_AM_MASTER messages from the master as well.

After slave finishes one self-test period, the master and the slave exchanges their roles.

C. Messages Sent Between Microcontrollers

Both MSP430 microcontrollers are connected by bidirectional duplex asynchronous serial line. The line is used for commands and state information messages transfer between microcontrollers.

In every state of operations, both microcontrollers send periodic messages to inform their counterparts that it is alive. The master sends I_AM_MASTER message while the slave sends I_AM_SLAVE message. If due to some reason the master or the slave switched itself to wrong mode (e.g. after unexpected reset by watchdog), the second microcontroller receives wrong message (e.g. slave receives I_AM_SLAVE from his counterpart). Microcontroller that detects such inconsistent state sends RESET_COMMAND to force reset on the other one and then reset herself.

Before each step of self-testing, the slave sends TEST_START message to the master to signal beginning of the test. After the self-test step finishes, the slave sends TEST_FINISHED message to the master with results of this test step. If TEST_FINISHED will not be received by the master until the defined deadline, the master supposes some fatal failure during slave testing procedure and begins slave resuscitation procedure.

Each event detected by the master (sending and/or receiving the message, unexpected state of the system etc.) is stored to the event log record by the master. Event log record is a dedicated area of FRAM memory. When satellite downlink communication is established, the satellite communication system will read the event log record and send it to Earth.

D. Resuscitation of The Dead Microcontroller

Main goal of the PilsenCube experimental payload is to prove the MSP430 microcontroller functionality in the LEO environment. Two microcontrollers on the payload board permanently checks each other. If one of microcontrollers fails, the second one has some ways to resuscitate it.

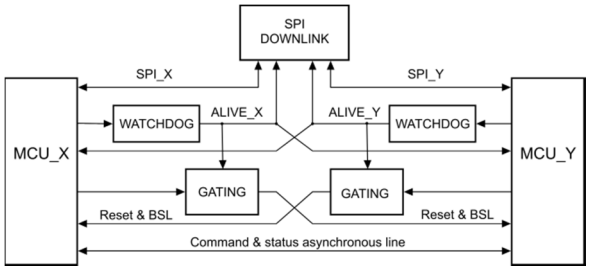

Figure 2. The PilsenCube picosatellite and the experimental payload block schematic.

Each of microcontroller checks proper operation of its counterpart by two ways:

- Checking the ALIVE_*x* signal from the second microcontroller. The ALIVE_*x* signals are generated by two external watchdog devices. Each watchdog device is serviced by one of microcontrollers. Latched watchdog output signal is sampled by the second microcontroller to detect possible malfunction of the first one.
- Checking state messages on serial interconnect line between both microcontrollers. Each microcontroller should receive periodic message I_AM_MASTER or I_AM_SLAVE sent by the second one. If the message is not received until defined timeout expires, the receiving microcontroller supposes its counterpart to be out of order.

If one of the microcontrollers detects malfunction of the second one, it has some means to try to resuscitate it.

In the first step, it will try to restart the second microcontroller by activating its reset input pin. If the malfunction of the ill microcontroller was caused by SEU, this step should bring it to the proper operational state. Reestablished ALIVE_*x* signal will indicate to the rescuer that the attempt was successful.

If the resetting attempt was not successful, there is the possibility that the malfunction was caused by soft error which modified program code stored in the FRAM memory. Each of the microcontrollers holds in its FRAM memory spare copy of the program code of the second one. Now the second step of the resuscitation process starts: By activating appropriate

pins, the healthy microcontroller will try to start the bootloader of the ill one. If the bootloader answers, the spare copy of the code will be written to the ill microcontrollers' memory and reset will be tried again.

The resuscitation attempts will be repeated periodically for the whole payload lifetime.

Control signals for resetting the microcontroller and for starting its bootloader are sent from GPIO pins of the second, perhaps healthy one. There is a risk that the failure of one microcontroller will permanently reset and therefore paralyze the second one. Therefore these dangerous lines are gated by ALIVE_*x* signals. If the respective microcontroller is not alive, the lines are forced to the inactive levels.

VI. SELF-TESTS OF THE MICROCONTROLLERS

In each instant, one of the microcontrollers program is in the master state while the second is in the slave state. The microcontroller in the master state does not perform any self-testing. Microcontroller in the slave state executes a set of self-test procedures, testing FRAM memory and some selected functional and peripheral modules. Due to the lack of the JTAG BSC, only some kind of functional testing can be executed. The self-test set is based on the TI's IEC60730 test library.

A. FRAM memory testing

MSP430 microcontroller used in experimental payload has 256 kB of FRAM memory. The memory can be used as program code storage as well as the data storage. Testing behavior of this memory in LEO conditions is one of important goals of the experiment.

The memory functionality is tested in three different ways:

- MSP430 FRAM memory controller has built in circuit which can correct single bit error and/or to detect uncorrectable multiple bits error. If some of those events occur, the corresponding status information in FRAM controller registers is set and interrupt signal is asserted. The payload software uses this information for continuous monitoring of memory health.
- In the second step the content of the memory is verified. In the first program startup procedure after new software had been written to the memory the CRC checksum is computed for each 1 kB block and stored in dedicated location of the memory. The memory checksums are re-computed and compared with stored values during periodic self-test.
- In the third step, FRAM memory is by-block tested by March-C algorithm [3]. A part of free memory area is used to temporarily store content of the tested memory block.

B. Other MSP430 function units testing

Besides the FRAM memory testing the microcontroller proceeds functional testing of selected on chip peripheral modules. To support self-testing, some external connections of unused serial input and output lines was added as well as connections of selected analog inputs to the external resistor network.

VII. CONCLUSION

The PilsenCube picosatellite should be deployed in the next year. We will publish results of the presented experiment as soon as they will be available.

If the MSP430 FRAM microcontroller will show capability to work in the LEO environment, it will bring possibility to apply this middle-class and easy to use microcontroller in a number of various experiments in small satellites.

ACKNOWLEDGMENT

This work was supported by Ministry of Education, Youth and Sports of the Czech Republic, Institutional support for long-term strategic development of research organizations.

REFERENCES

[1] D. Sinclair and J. Dyer, "Radiation Effects and COTS Parts in SmallSats," in Proceedings of the AIAA/USU Conference on Small Satellites, SSC13-IV-3, 2013.

[2] T. C. MacLeod, W. H. Sims, K. A. Varnavas, F. D. Ho," Results from on-Orbit Testing of the Fram Memory Test Experiment on the Fastsat Micro-Satellite," Integrated Ferroelectrics, Volume 132, 2012, Pages 88-98.

[3] [3] M. Mamatha, M. Muralidhar: Memory Testing using March C Algorithm. International Journal of VLSI System Design and Communication Systems, Vol. 02, Issue 07, October 2014, pp. 0512-0517.

[4] Fiala, P., Voborník, A. Embedded microcontroller system for PilsenCUBE picosatellite. In Proceedings of the 2013 IEEE 16th International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS). Brno 2013. s. 131-134. ISBN: 978-1-4673-6136-1

[5] Vobornik, A., Veřtát, I. Radiační odolnost pikosatelitů. Slaboproudý obzor, 2011, roč. 67, č. 4, s. 1-3. ISSN: 0037-668X

[6] Veřtát, I., Voborník, A. Efficient and Reliable Solar Panels for Small CubeSat Picosatellites. INTERNATIONAL JOURNAL OF PHOTOENERGY, 2014, roč. 2014, č. June, s. 1-8. ISSN: 1110-662X

[7] Veřtát, I., Linhart, R., Masopust, J., Voborník, A., Dudáček, L. Earth's thermal radiation sensors for attitude determination systems of small satellites. Contributions of the Astronomical Observatory Sklanaté Pleso, 2017, roč. 47, č. 2, s. 157-164. ISSN: 1335-1842

[8] Veřtát, I., Linhart, R., Dudáček, L., Dániel, V., Svoboda, P. Autonomous and semi-autonomous radio commanding of VZLUSAT-1 nanosatellite from ground control station in Pilsen. In International Conference on Applied Electronics (AE 2017) : /proceedings/. Pilsen: University of West Bohemia, 2017. s. 263-268. ISBN: 978-80-261-0641-8 , ISSN: 1803-7232

[9] VZLUSAT Pilsen Ground Station data. https://www.pilsencube.zcu.cz/vzlusat1/ . (Accessed on 07/04/2018).