STUDENTS' SPACE ASSOCIATION

THE FACULTY OF POWER AND AERONAUTICAL ENGINEERING
WARSAW UNIVERSITY OF TECHNOLOGY

PW-SAT2

PRELIMINARY REQUIREMENTS REVIEW

On-Board Computer

Phase A of PW-Sat2 project

1.0 EN

pw-sat.pl

2014-05-08

Abstract

The following paper is a part of Phase A Summary of student satellite project PW-Sat2. The document presents a first version of a satellite's on-board computer (OBC).

The document is published as a part of:

PW-Sat2 – Preliminary Requirements Review

REVISIONS

Version	Date	Changes	Responsible
	2014-01-12	Creation of this document. Added "OBC in CubeSat missions".	Piotr Kuligowski
	2014-01-13	Added: "Requirements for PW-Sat2's OBC", "Description of PW-Sat2 subsystems"	Piotr Kuligowski
	2014-02-21	Added: "Selection of components for the OBC0 1v0 prototype"	Piotr Kuligowski
0.9 EN	2014-05-08	English translation	Błażej Kula
1.0 EN	2014-05-08	Editing	Dominik Roszkowski
1.0.1 EN	2014-07-02	Small editorial changes	Dominik Roszkowski

This document is also available in Polish.

TABLE OF CONTENTS

CONCEPTS AND ABBREVIATIONS

ADCS – Attitude Determination and Control System.

ARM – Advanced RISC Machine – company name and architecture type.

AVR – ATMEL's microcontroller family.

BMP – raster graphics image file format.

CDMS – Command and Data Monitoring System.

COMM – communication subsystem on-board satellite.

DMA – Direct Memory Access.

ECC – Error Correction codes – correction codes which enable limited repair of damaged data.

EEPROM – Electrically-Erasable Programmable Read-Only.

EPS – Electrical Power System –power subsystem on-board satellite.

Flash –type of EEPROM memory. NOR Flash and NAND Flash marked out.

FPGA– Field Programmable Gate Array .

FRAM– Ferroelectric RAM.

GIF– Graphics Interchange Format – method of lossless compression of series of raster graphics images.

I²C – Inter-Integrated Circuit –two-wire serial bus.

JPG – raster graphics images lossy compression method.

JTAG– Joint Test Action Group – interface for programmable logic arrays testing.

LEO – Low Earth Orbit.

MCU –single chip microcontroller.

OBC – On-Board Computer.

OBC0 – main computer on-board satellite.

OBC1 – Reserve computer which replaces OBC0 in case of its malfunction.

OS – Operating System.

PCB – Printed Circuit Board.

PIC – Microchip's microcontroller family.

PLD – Payload.

PNG– Portable Network Graphics – a raster graphics file format that supports lossless data compression.

P-POD– Poly-PicoSatellite Orbital Deployer.

RAM – Random Access Memory.

SPI – Serial Peripheral Interface.

SRAM – Static Random Access Memory.

RAID - Redundant Array of Independent Disks.

RTOS – Real Time Operating System.

USART – Universal Synchronous and Asynchronous Receiver and Transmitter.

µC – microcontroller.

1 INTRODUCTION

OBC team's task is to design and build an on-board computer for the PW-Sat2 satellite in the form of PC-104 single-board computer. Module will function in vacuum environment with large temperature gradient and increased ionizing radiation

This document contains the method of selection of an architecture for PW-Sat2's OBC, based on requirements of the mission and analyses of other Cubesat satellite's OBCs. The most optimal redundancy of OBC's basic blocks was selected together with systems to prevent uncontrolled looping of currently processed program. Parts for the first prototype of OBC were selected.

2 OBC IN CUBESATS' MISSIONS

Choice of microcontroller/microprocessor used in OBC depends on tasks which the satellite is designed to perform. Other computers are used in satellites performing imaging tasks and other when satellite is used as a transponder.

If there is a camera on-board satellite, images from it need to be received (usually through parallel output bus) and saved to a Flash memory. In case where data aren't compressed, they can be compressed with an on-board computer. Compression can be done more economically with an FPGA circuit. Microcontroller requires more energy to compress data while FPGA requires more work to create the circuit itself (even though we are going to use libraries for these circuits).

When satellite is tasked only to receive and broadcast data, without interference with data structure, and only with frequency change, 8-bit microcontroller is sufficient. Additionally it can be used as power system's controller at the same time. Second microcontroller, identical to the first one, can be used as a redundant system.

On-board computers are chosen based on their "flights" histories. If given computer was on orbit many times and functioned for at least a year, there is high probability that when we buy the same computer it will function at least a year.

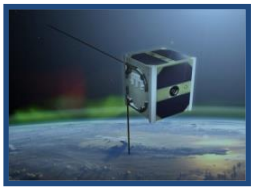
2.1 EXAMPLES OF CUBESAT COMPUTER'S CONSTRUCTIONS

No	Mission	Launch date and orbit	µC in OBC	Memory	Operating system	Mission status
1	AAUSat	2003-06-30 14:15 650 km – Sun-synchronous	**Siemens C161-RI (8051)**	External RAM: 4MB Ext. Flash: 128kB Ext. PROM: 128kB	no	**Mission failed.** Deactivated on 22 September 2003.
2	QuakeSat	2003-06-30 14:15 650 km – Sun-synchronous	**Prometheus PC-104 (ZFx86)**, reduce 100 MHz to 66 MHz to reduce power,	External SRAM: 32MB Flash disk: 192MB	Red Hat 9 Linux	**Mission successful**

			PIC **16F628-20P** for watchdog.			
3	GeneSat-1	2006-12-16 12:00 460km, 40.5 degree inclination	**PIC18** microchips as OBC for all onboard monitoring and control	External Flash	no	**Primary mission complete**. GeneSat-1 re-entered the Earth's atmosphere on the August 4, 2010.
4	CanX-1	2003-06-30 14:15 650 km – Sun-synchronous	Atmel **ARM7** OBC operates at 3.3 V, consumes 0.4 W at a speed of 40 MHz.	External SRAM: 512kB storage of bulk data: 32MB Boot ROM is a 128 kB.	eCos	No signal from spacecraft. **Mission failed.**
5	QbX1 & QbX2	2010-12-08 300 km	**SI Labs 8051**	?	?	**Mission successful**
6	M-Cubed	2011-10-28	Stamp9G20 from Taskit. 400MHz **ARM9** core.	SDRAM: 54M NAND Flash: 128MB	RTLinux	**Unclear.**
7	AubieSat-1	2011-10-25 452km x 755km	**ATMEGA 128**	External RAM	Real Time OS	**Mission successful.** As of April 2013, the satellite continues to transmit.
8	AAUSat-3	2013-02-25	**ARM7**	?	FreeRTOS	**Active**
9	ESTCube-1	2013-05-7 670km	**ARM Cortex-M3 STM32F103VET 6** at a speed of 72MHz	External Flash: 16MB SPI External FRAM: 256kB	FreeRTOS	**Active**
10	Cute-1.7 + APD II	2006-02-21 21:28:00 UTC 299 x 712km	**ARMV4I** 400MHz	External RAM: 32MB SD Card: 128MB	Microsoft Windows CE.NET	**Failed while in orbit.**
11	SwissCube -1	2009-12-23 6:21 UTC	ATMEL **AT91M558800A**	External SRAM: 1MB 2MB dev. software	eCOS	**Active**

		726 x 752km Sun-synchonous	and **MSP430F1611** is used to do the SPI to I²C conversion	flash 8MB storage flash		

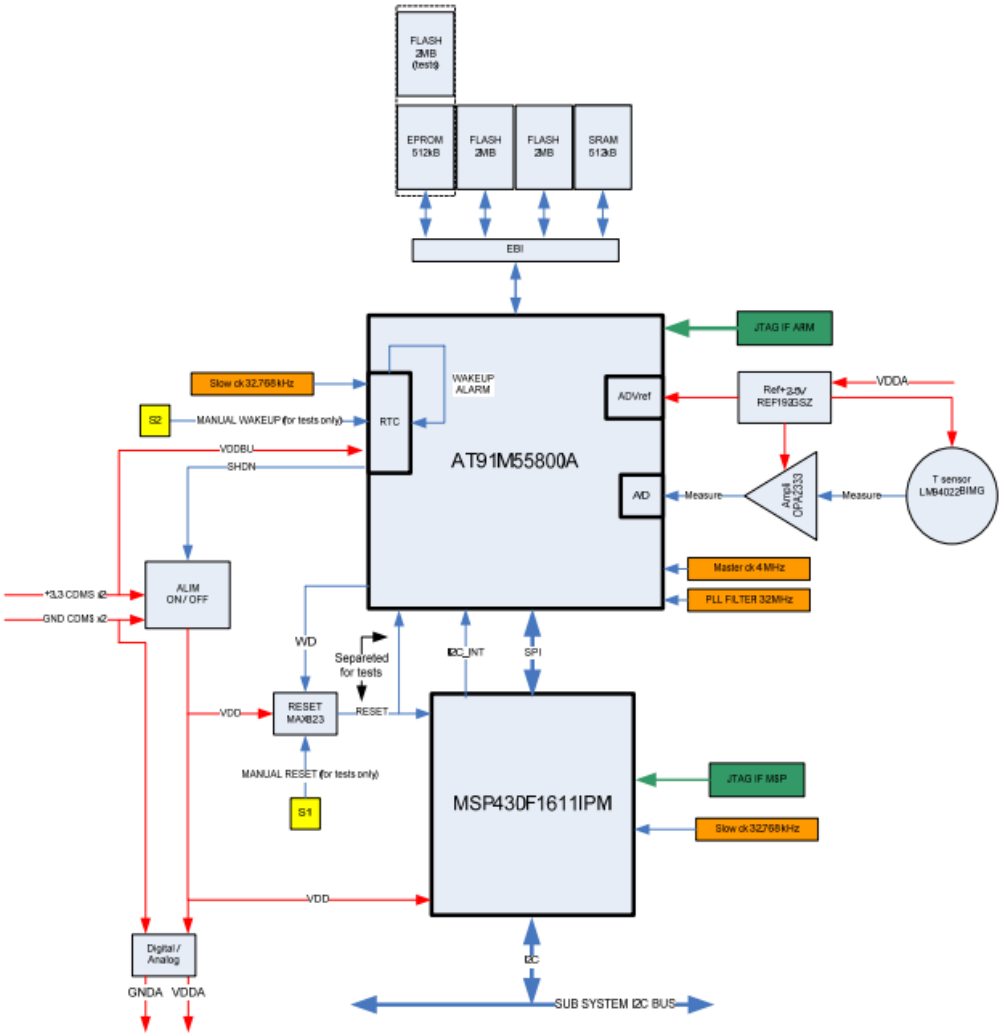
2.2 SwissCube's on-board computer



SwissCube (http://swisscube.epfl.ch) is a Swiss satellite which follows Cubesat 1U standard. It was built for an airglow (nightglow) observation in infrared and also it was built as a technology demonstrator. SwissCube's team shared documentation of the project on their website (http://ctsgepc7.epfl.ch). It is one of the best documentations of this type available online. The following is the analysis of SwissCube's CDMS structure based on available documentation.

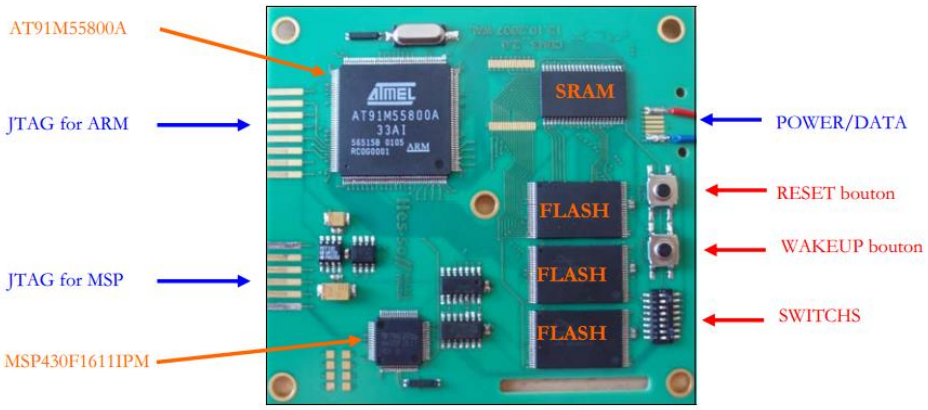
2.2.1 Construction of the SwissCube's on-board computer - phase C

Prototype with two microcontrollers: MSP430DF1611IPM and AT91M55800A (ARM7TDMI), together with external storages EPROM 512kB (for program's code), Flash 2x2MB and SRAM 512kB was built. Also watchdog system (MAX823) and thermometer was placed externally. The block diagram is shown below.



More simple MSP430 functioned as SPI<->I²C translator, because AT91M55800 doesn't have I²C interface. Obtained this way I²C functioned as an on-board data bus for communication between subsystems and CDMS.

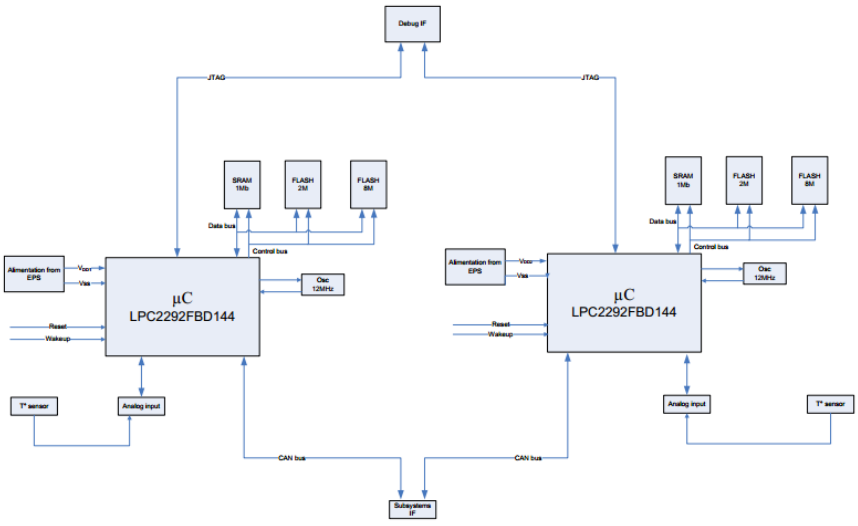
Engineering Qualification Model - CDMS prototype for SwissCube in phase C:



Space available on PCB board was utilized suboptimally. Two JTAG edge connectors take too much space. Tantalum capacitors aren't optimal choice for using in places which require high reliability.

2.2.2 CONSTRUCTION OF THE SWISSCUBE ON-BOARD COMPUTER – PHASE D

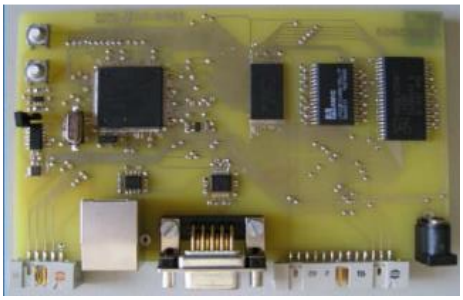
In phase D the idea of the on-board computer was changed. Microcontroller was swapped for LPC2292FBD144 (ARM7TDMI). Second computer, identical to the first one was added as a redundant one. The block diagram of SwissCube's phase D on-board computer is presented below:



In this version memory capacity was increased: SRAM to 1MB and Flash to 2+8MB.

2.2.3 SUMMARY OF SWISSCUBE'S CDMS ANALYSIS

Ultimately version with ARM7TDMI+MSP430 microcontroller was used in the final model. eCos was used as an operating system.



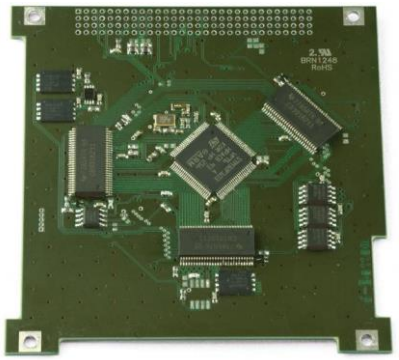
2.3 ESTCUBE-1'S ON-BOARD COMPUTER

ESTCube-1 is first Estonian satellite. It was placed on 670 km orbit in May 2013. As of today (January 2014) it still functions. It's main objectivewas educational. It also tested functioning of an electrical sail. It was equipped with color VGA camera. Mission was successful. On the project's website http://www.estcube.eu interesting information concerning some subsystems.

2.3.1 CONSTRUCTION OF THE ESTCUBE-1 CDHS

Estonians used STM32F103VFT6 microcontroller in the on-board computer. It is up-to-date 32-bit ARM Cortex-M3. Three external Flash 16MB SPI storages for camera data storage together with five external FRAM storages with SPI interface to storage data settings, system files and OS image.

No.	Device	Feature	Value
2	STM32F103VFT	Flash	768 KB
		SRAM	96 KB
		Clock	.. 72 MHz
5	FM25V20	SPI FRAM	256 KB (1280 KB in total)
		SPI Clock	.. 20 MHz
3	S25FL128P	SPI Flash	16 MB (48 MB in total)
		SPI Clock	.. 20 MHz
1	DS3234	SPI RTC	



Internal program's Flash memory capacity is 768kB. It is a lot. Additional program's NOR external Flash memory becomes unnecessary. It allows to reduce the number of connections on the computer's board.

Everything runs under FreeRTOS operating system.

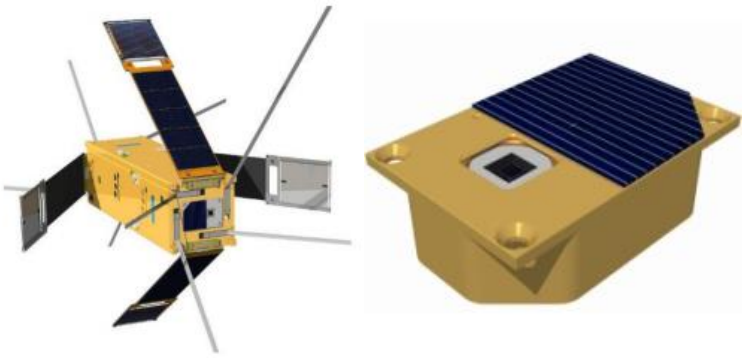
2.3.2 SUMMARY OF ESTCUBE-1'S CDHS ANALYSIS

Success of this mission proves that it is possible to use STM32 microcontrollers on orbit. Despite the ionizing radiation is increased and structures deterioration is greater than on Earth the system can still function for one year (as in ESTCube-1).

There is huge quantity of peripheral devices for STM32 microcontrollers. There is no need to use external SPI<->I²Cconverter (as in SwissCube). STM32 microcontrollers are good choice in present-day Cubesats which function on LEO doesn't exceed one year.

2.4 DELFI-C³'S ON-BOARD COMPUTER

Delfi-C³is first Dutch student's satellite (http://www.delfic3.tudelft.nl). It tested functioning of the elastic solar panels and autonomous Sun sensors. Sun sensors had autonomous power source (small solar panels) and were connected to the computer wirelessly on 915MHz frequency. Passive ADCS was built using permanent magnets. One of the experiment was to test the satellite's function as a linear radio transponder.

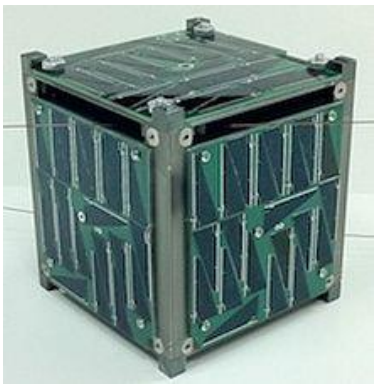


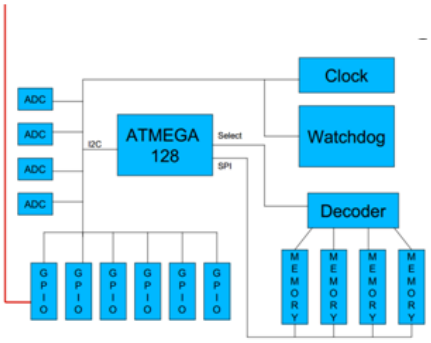
Commercial FM-430 Pumpkin kit based on MSP430 microcontroller(Texas Instruments) was used. I²C is the system bus. There were certain anomalies in I²Cfunctioning which hindered the mission execution, although ultimately the mission was successful.

MSP-430 is one of the most popular microcontrollers used in Cubesats. It is compatible with Von Neuman architecture (program's code and data are in the same memory). Characteristic feature is energy efficiency. There are versions available with FRAM memory.

2.5 AUBIESAT-1'S ON-BOARD COMPUTER

AubieSat-1 was launched in October 2011. Its task was to demonstrate technology. Mission was successful. It was still possible to connect to the satellite in April 2013.

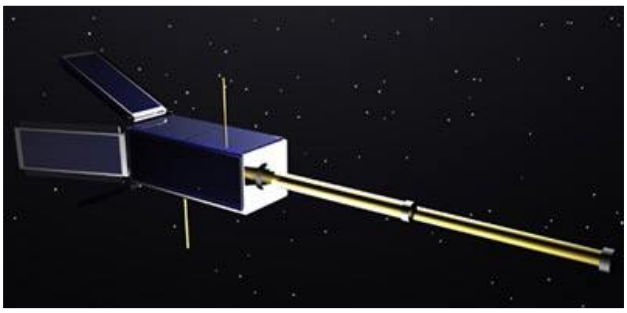




It was running on ATMega128 microcontroller with RTOS operating system. Data were stored on Flash memory with SPI interface.

2.6 QUAKESAT'S ON-BOARD COMPUTER

QuakeSat was launched in June 2003. It main objective was detecting low-frequency electromagnetic signals, so it could help detect earthquakes. Mission was successful.



Main QuakeSat's computer was based on x86 processor which worked with 66MHz clock rate (Prometheus PC-104 ZFx86). At first its clock rate was 100MHz but it was lowed to 66MHz to reduce the amount of emitted heat. 32MB of RAM together with 192MB flash disk were installed. Everything was working under Red Hat 9 Linux operating system. Microcontroller PIC16F628-20P functioned as a hardware watchdog.

2.7 SUMMARY OF OBCS ANALYSIS IN VARIOUS CUBESAT MISSION

Based on this analysis it can be concluded that it is common to use commercially available microcontrollers to build on-board computers for satellites working on LEO. Basic requirement for microcontroller is that it has wide operating temperature, which is fulfilled by most

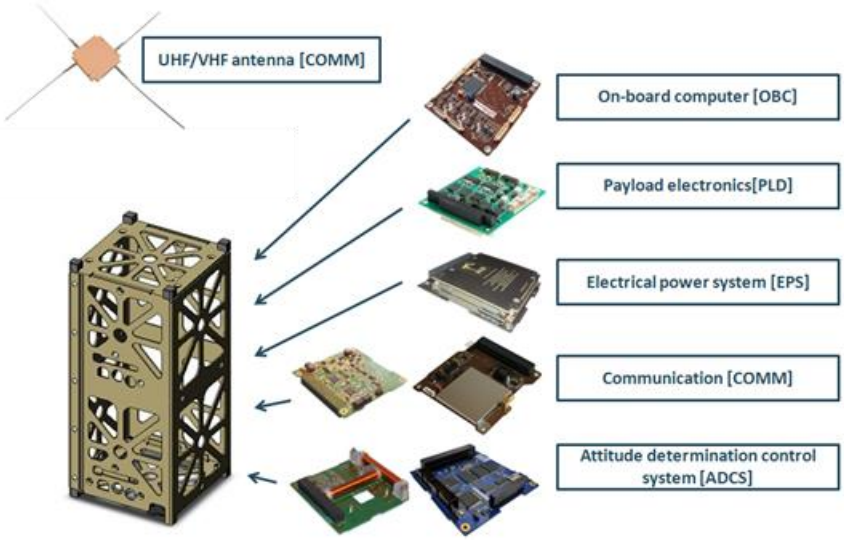
comercially available units. Limitation to functioning time on orbit is ionizing radiation which leads to structure deterioration.

ESTCube-1 shows that present-day 32-bit microcontrollers such as STM32 can efficiently function on LEO for at least a year. Due to their higher computing power they can do more things than, for example 16-bit MSP-430 microcontroller.

AubieSat-1 shows that microcontrollers from AVR family can work for at least 2 years on LEO. They are very simple to programme although they don't have high computing power. They are suitable for less complicated tasks such as controling of power system

3 DESCRIPTION OF PW-SAT2 SUBSYSTEMS

The main task of the PW-Sat2 satellite is to test the deorbitation system. This system functions by increasing the aerodynamic drag. At low orbit there is vestigial atmosphere, which will allow it. The effect of the drag is lowering the orbit and at some point entering the lower parts of atmosphere and burning.



Second task is to test Sun sensor which consists of 4 small solar panels. Current is measured on each panel. Based on these measurementsand tables from Flash memory Sun sensor's microcontroller determines satellite's position. Sun sensor is connected to the on-board computer with I^2C bus.

Power system receives energy from solar panels and stores it in batteries, controls turning on and off the power and reacts to critical situations. Switches are controlled based on commands from the on-board computer.

Everything is controlled by on-board computer. It processes commands received from Earth, monitors power budget, automatically does tasks based on built-in schedule and receives data from cameras.

Communication subsystem allows the connection with ground station, sending stored data and receiving commands.

ADCS, Attitude Determination and Control System. Active magnetic system will allow for limited orientation.

4 REQUIREMENTS FOR PW-SAT2'S OBC

OBC team's task is to design and build an On-board Computer (OBC) for the PW-Sat2 satellite in the form of PC-104 single-board computer. Module will function in vacuum environment with large temperature gradient (-40°C to +60°C) and increased ionizing radiation.

4.1 BASIC TASKS FOR THE ON-BOARD COMPUTER

a) Communication with other satellite subsystems via I2C and SPI interface. System and payload(experiment) bus are separated.
b) Be the timer of the mission from the moment of ejection from the P-POD and turning on power.
c) Performing scheduled tasks - built-in tasks list to perform during mission.
d) Enabling performing any task based on command from Earth, before time provided within task list.
e) Performing simple scripts sent from Earth.
f) Monitoring of the power budget and issuing commands to EPS which turns on/off power switches.
g) Receiving data from camera CAM2 as compressed series of VGA photos (5-10 frames/s) or single VGA image. Received data are to be saved in Flash memory.
h) Handling parallel NOR Flash memory for images data, Static RAM memory (minimum 1MB operating memory), serial Flash memory (measurements of the currents, voltages and temperatures onboard satellite) and small block of FRAM memory (ferrite-core memory, for storage of essential configurations and the map of damaged memory cells).
i) Computer needs to be overseen by a hardware watchdog.
j) To the reset line needs to be connected hardware reset.

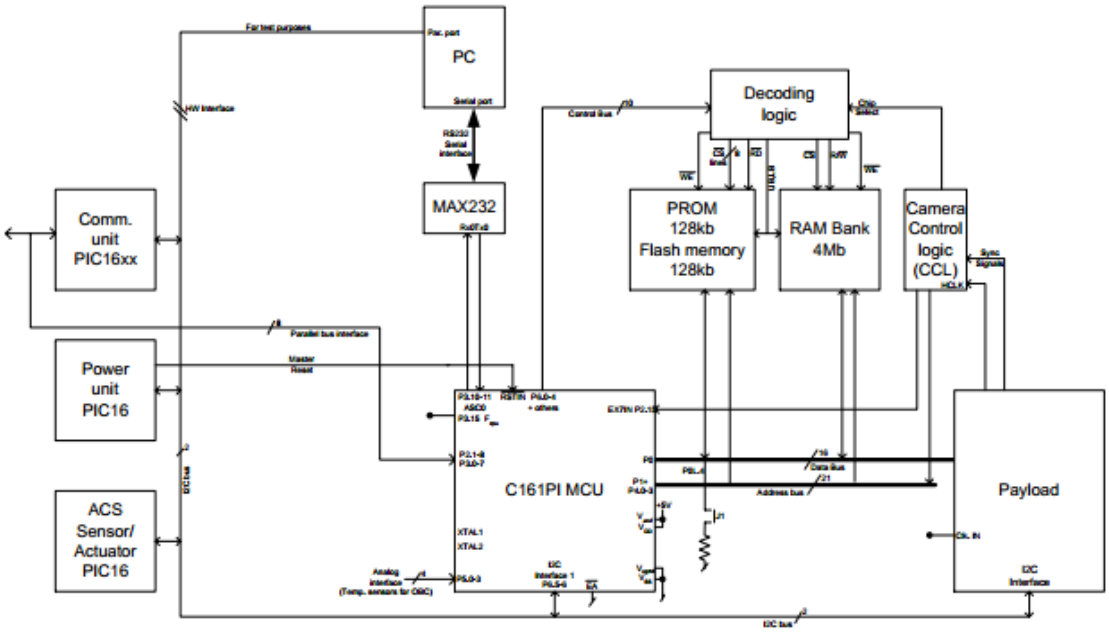
4.2 VARIOUS OBC ARCHITECTURES

Based on analysis of Cubesat missions done in previous subsection it is clear that there are several basic OBC architectures. They are presented below, along with the conclusions.

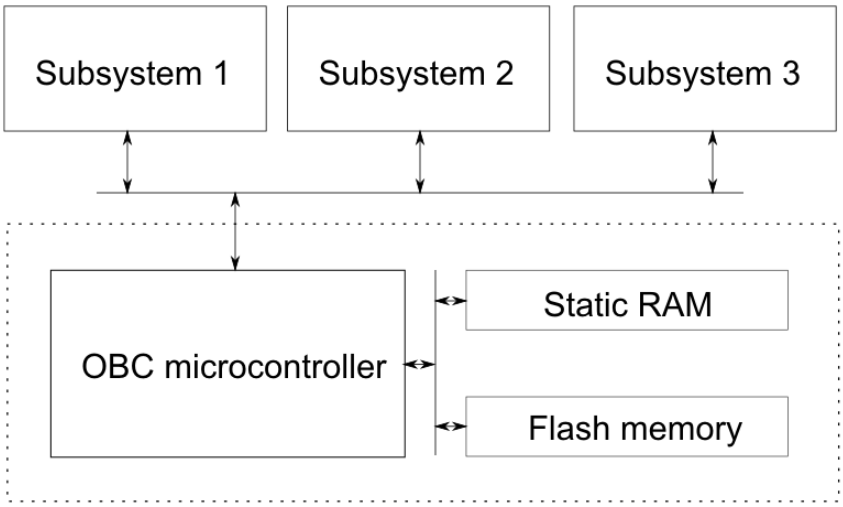
4.2.1 SINGLE, CENTRAL MICROCONTROLLER

A computer consists of one main microcontroller which communicate with satellite's subsystems. Primary flaw in this solution is lack of redundancy. Damage of main processor prevents from continuing the mission. Example of usage of this architecture is AAUSat.

The mission was a failure because stable communication wasn't established. It isn't clear why this happened. Flowchart of AAUSat:



Simplified flowchart, introducing the idea of "one main processor":



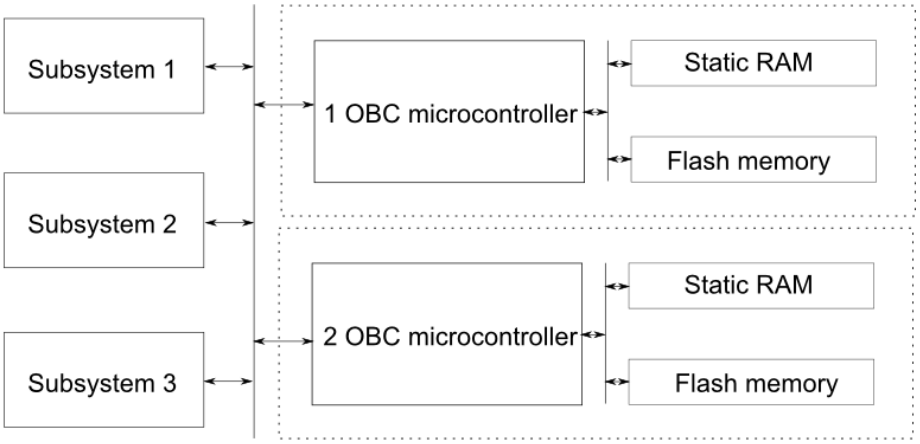
The only advantage of this solution is very easy programming.

4.2.2 TWO IDENTICAL, REDUNDANT MICROCONTROLLERS

System is built from 2 identical, independent and redundant computers. When one malfunctions, second (identical) will take its place. In the role of "arbitrator" deciding which processor has control at any given time, power system can be used. It polls given computer and if it doesn't answer then it gives control to the second one.

This solution was used in SwissCube prototype in phase D. Ultimately another model made it to the orbit.

Simplified flowchart introducing the idea of connecting two independent microcontrollers:



A disadvantage of this solution is that in two identical microcontrollers structure's degradation caused by ionizing radiation proceeds at similar rate. So when one microcontroller malfunctions due to this, there is high probability that the second one will be damaged in similar degree. Another problem is the space occupied on the board. Solution to this problem may be in the next subsection.

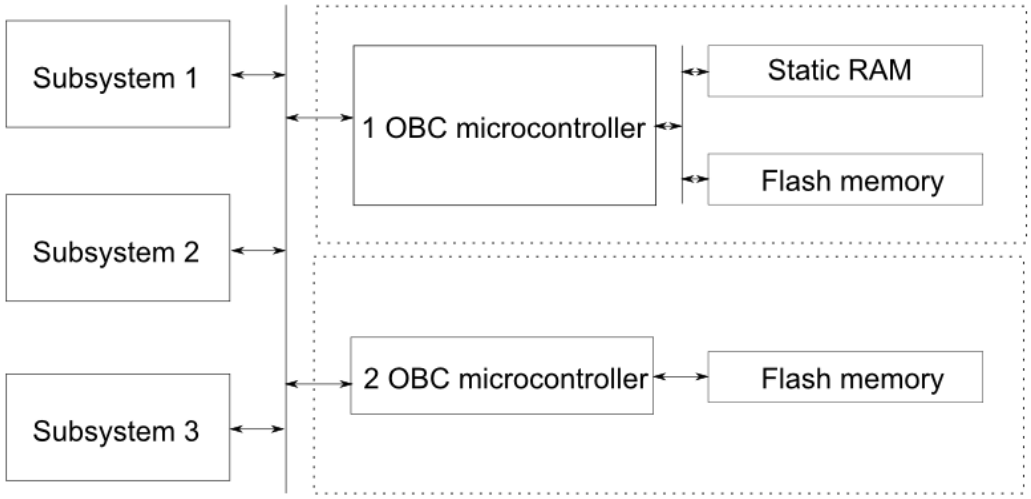
4.2.3 TWO DIFFERENT, REDUNDANT MICROCONTROLLERS

In a solution where there are two identical processors there is a chance that when one is damaged with ionizing radiation the second one can be damaged in similar degree.

We can build a computer using two different microcontrollers:

a) First one, main, very advanced, with large scale of integration - needed for cameras images processing, complicated calculations, sending large amount of data.

b) Second one, reserve, with much smaller level of integration - needed in case of malfunction of the main one. He would be able to replace the main processor in the simpler tasks.



On the scheme above one can find the main microcontroller (number 1) and the secondary (redundant) microcontroller (number 2). No 1 takes care of processing images and sending images between cameras and OBC

Structure degradation in two different levels of integration will causes different damage in the same time. More advanced processor will malfunction much earlier. Presence of a less advanced microcontroller allows mission to continue although in limited way. Storage of many images or their processing can't be done without advanced processor. It may be possible to send an image to Earth "on the fly" and it issuing commands to subsystems will still be possible.

4.3 SELECTION OF THE ARCHITECTURE FOR PW-SAT2'S OBC

The following table contains advantages and disadvantages of three different architectures, described wider in previous sections.

	Disadvantages	Advantages
One main microcontroller	- In case of malfunction of the main microcontroller mission cannot be continued.	- Programming is very easy. - PCB board is smaller than in other solutions.
Two identical, redundant microcontrollers	- In microcontrollers with similar level of integration structure degradation proceeds at similar rate. - Elements on PCB board take up twice that much space when compared to solution with only	- When one of the microcontrollers overheats, second one can take its tasks with the same configuration. - Software can be identical, EPS would make decision which one is controlling the satellite..

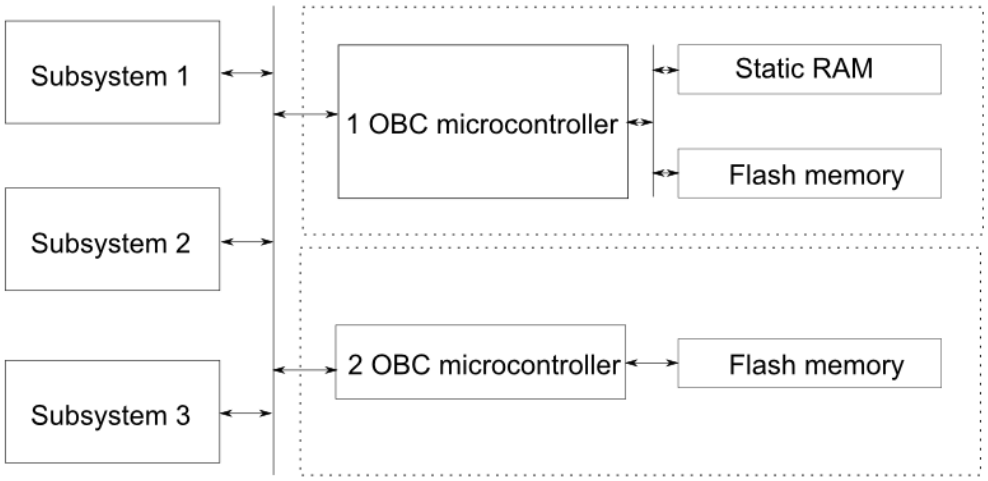
	one, main microcontroller. - ESP needs to be arbitrator which determines which microcontroller works at a given moment.	
Two different, redundant microcontrollers	- Software for the first one and the second one must be different (at least for hardware drivers).	- Degradation process proceeds at different rate. Microcontroller with smaller level of integration will work longer. - Space taken up on PCB board is a compromise between solutions above.

Due to limited space available onboard PW-Sat2 we can't use the solution with two identical microcontrollers. Additionally there is a problem mentioned above with similar rate of structure degradation. We also can't choose solution with only one, main microcontroller. In this case OBC would be without redundancy.Subsystem as important as OBC requires redundancy of basic blocks. Therefore for PW-Sat2 satellite an architecture with two different, independent microcontrollers was chosen. One of the microcontrollers will be much more advanced which will allow receiving, processing, saving and sending of many images. Less advanced microcontroller will be reserve with smaller level of integration. It will allow longer functioning time on orbit.

4.4 NECESSARY OBC'S MEMORY BLOCK

Storage devices can be divided into: non-volatile memory and volatile memory. Volatile memory is used for data storage during functioning of the operating system and data are lost after disconnection of the power supply. In non-volatile memory data aren't lost after the power supply is disconnected. PW-Sat2's on-board computer needs different memory blocks for saving results of current tasks or saving the images.

In previous analyses architecture consisting of two different microcontrollers (one advanced and fast, and the other one slower although less vulnerable to ionizing radiation) was chosen. Simplifieddiagram:



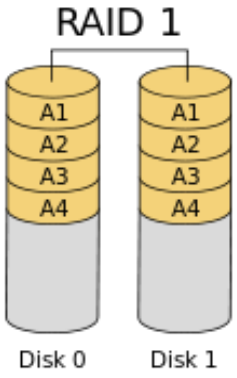
This diagram presents the idea of connecting two different microcontrollers. In this section it will be extended with memory blocks.

4.4.1 MEMORY FOR IMAGES AND FILMS FROM CAMERA

For saving images and films non-volatile memory is necessary, for example NOR Flash or NAND Flash. It must allow for saving at least 20 images and 10 films from CAM2 camera 30s long each, together with 10MB payload data. To ensure the highest possible data transfer rate it must be parallel memory, operated directly by main microcontroller.

4.4.2 MEMORY FOR SYSTEM SETTINGS

Non-volatile memory is necessary here as well although different than this used for storage of images and films. System's setting data(something like hardware register), maps of damaged RAM memory cells, constants etc. will be saved in it. Based on this much higher resistance for ionizing radiation will be needed, which FRAM memory fulfills.



In order to boost the reliability two identical memory blocks can be used for settings data saving. Both blocks would have the same data, just like in RAID 1 hard disks array.

This memory needs to be accessed also by the redundant microcontroller. This way, after malfunction of the main computer it could access settings data and continue interrupted tasks.

4.4.3 RAM MEMORY

For image processing, packaging and sending data, much RAM memory is needed. RAM memory is used for storage of current results where main microcontroller uses it directly. High data transfer rate is required and space for saving at least one uncompressed image from any camera. For operating this memory, microcontroller with embedded Static RAM or Dynamic RAM memory controller is needed.

4.4.4 MEASUREMENTS MEMORY

In the EPS current and voltage on every power line is measured. Exceeding expected power consumption by a subsystem may cause damage of other systems. Constant monitoring and reacting can prevent this. From time to time system tests will be carried. Results need to be saved in non-volatile NOR Flash or NAND Flash memory. During each communication session measurements data can be sent to Earth and in case of malfunction we can react accordingly.

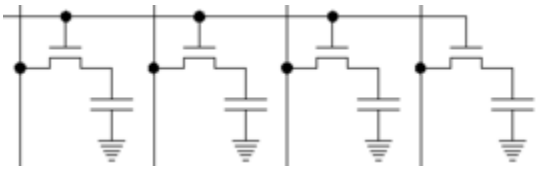
4.5 SELECTION OF MEMORY BLOCKS FOR OBC

For storage of current tasks's results volatile memory is commonly used (data is lost after power is lost). They are very fast and they can "keep up" with the processor which does computing. These features has static RAM memory, that is SRAM (static random-access memory).

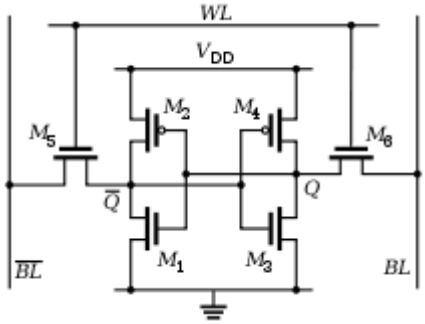
For permanent data saving we need to use non-volatile memory, e.g. Flash memory. Storage devices of this type have bigger memory space than main memory but they are also much slower. They are good at saving data on the same principle as hard disks in PC do. We are saving large amounts of data there and access it after some time.

4.5.1 DYNAMIC AND STATIC RAM MEMORY

DRAM is a type of volatile memory with random access (dynamic random-access memory). DRAM is usually arranged in a rectangular array (rows and columns) of charge storage cells consisting of one capacitor and transistor per data bit. Capacitors are storage cells. After some time capacitors are discharged and information is lost. To prevent this memory is periodically refreshed (through reading and writing to it again). It is widely used in PCs as primary storage. Its advantage is low price.



It is not used in space environment due to its failure rate.



Generally static RAM memory is used. SRAM is a type of volatile memory with random access. It uses bistable latching circuitry to store each bit. Latches are storage cells. After setting one of two states it stores it until the power is disconnected. It doesn't require memory refreshing and is faster than DRAM so it is used as processors cache memory. Although it is more complicated and more expensive.

SRAM memory will be used in PW-Sat2 for saving current computing data (e.g. image processing or data packing for COMM to send to Earth).

4.5.2 SELECTION OF CAPACITY OF RAM MEMORY

Image processing requires some minimal SRAM memory capacity. First let us consider loading uncompressed image from VGA camera (640x480pix) in RGB with 8-bit precision for every color:

- For each pixel there are 24 bits because 8bit+8bit+8bit=24bit (R+G+B) ,
- Number of pixels on VGA array: 640x480 = 307200 pixels, that is ~0,31 Mpix.
- Number of bits: 307200*24=7372800 bits
- Number of bytes: 7372800/8=921600, that is 900kB or ~0,88MB for one VGA image.

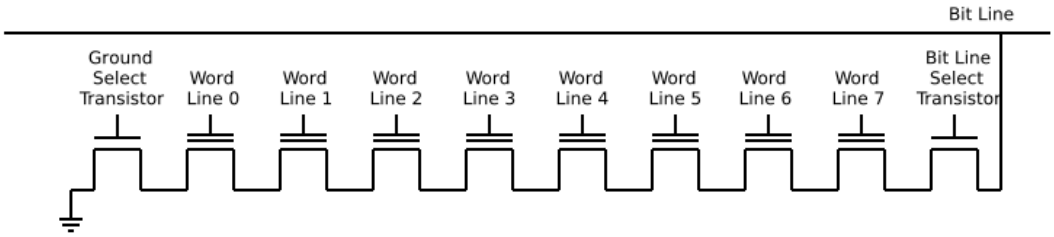
One uncompressed VGA image takes about 0,88MB in SRAM memory.

Even if the images from cameras will be compressed with JPG, SRAM memory will be chosen to the worst case scenario: uncompressed image. If there will be VGA camera onboard satellite, than minimal capacity of SRAM memory needs to be at least 1MB. SRAM memory with a capacity of 1 MB is avaible.

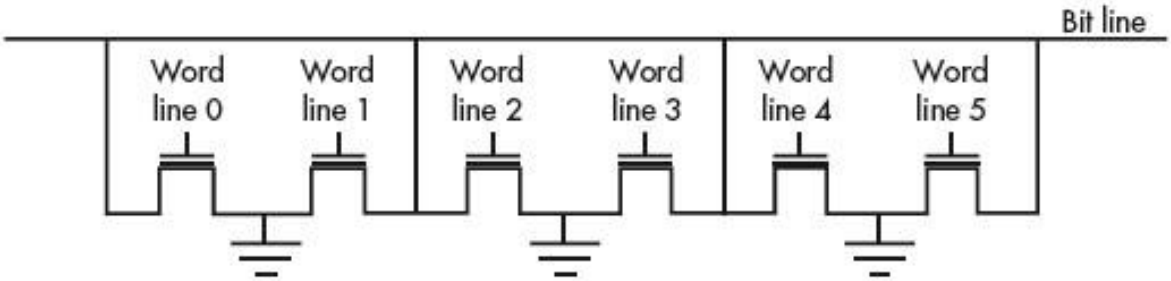
4.5.3 FLASH MEMORY FOR IMAGES, FILMS AND PAYLOAD

Flash is a non-volatile memory. It is used for holding large amounts of data, which aren't lost when power is disconnected. There are two types of Flash memory: NAND Flash and NOR Flash. They differ in internal structure.

NAND Flash



NOR Flash



In NOR Flash we have direct access to every memory cell and in NAND Flash there is a sequential access. Sequential access requires loading to RAM memory larger block of data from NAND Flash. NAND Flash technology is used in memory cards, mobile phones, PenDrives etc. NAND Flash requires more operations and therefore requires more energy than NOR Flash. In NOR Flash memory we are reading only this what we want. We have a shorter time access to data. Disadvantage of this memory is its high price and small memory capability - up to 512MB. In the satellite's on-board computer the power consumption during data reading is important. In this regard NOR Flash memory is a much better solution. Therefore for holding images, films and payload data onboard PW-Sat2 NOR Flash memory will be used.

4.5.4 SIZE OF ONE JPG IMAGE FROM CAM2 CAMERA

CAM2 camera onboard PW-Sat2 records the opening of the deorbiting structure. Its second objective is to photograph Earth's surface. Unfortunately in this kind of mission it isn't possible to send uncompressed images directly to the ground station. Communication session each time

will take the maximum of 15 minutes and transmission rate will be approximately 10kbps (UHF/VHF). Image needs to be compressed in manner that 5 images can be send in one communication session. In one session we can send maximum of 1MB of data, so the size of one image can't exceed 200kB.

Let us determine the size of one image: uncompressed and compressed with JPG algorithm in various qualities. Let us assume that 0,3 Mpix camera with 8-bit for each RGB color was chosen, that is 24 bits per pixel was chosen as the CAM2. For analysis we will use the photo of the MIR station taken from orbit (~400 km). Similar images could be taken by PW-Sat2. Image was scaled to the size of approximately 0,3 Mpix and was saved as a bitmap. Its size was 1MB. Compression rate is a reference to this number. Fragments of the image above was added to the analysis to show the impact of JPG compression in image's details.

The reference portion of the image, uncompressed photo of MIR station:

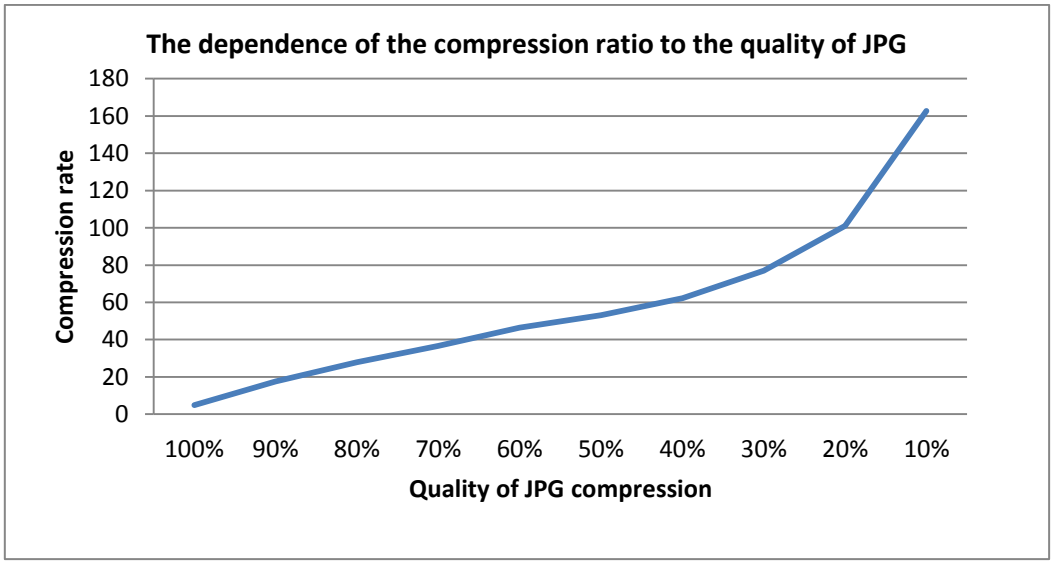
BMP parameters	Size of file	Portion of the image
R8bit, G8bit, B8bit	1MB	

The following table containts outcome of the analysis for JPG compression with different quality:

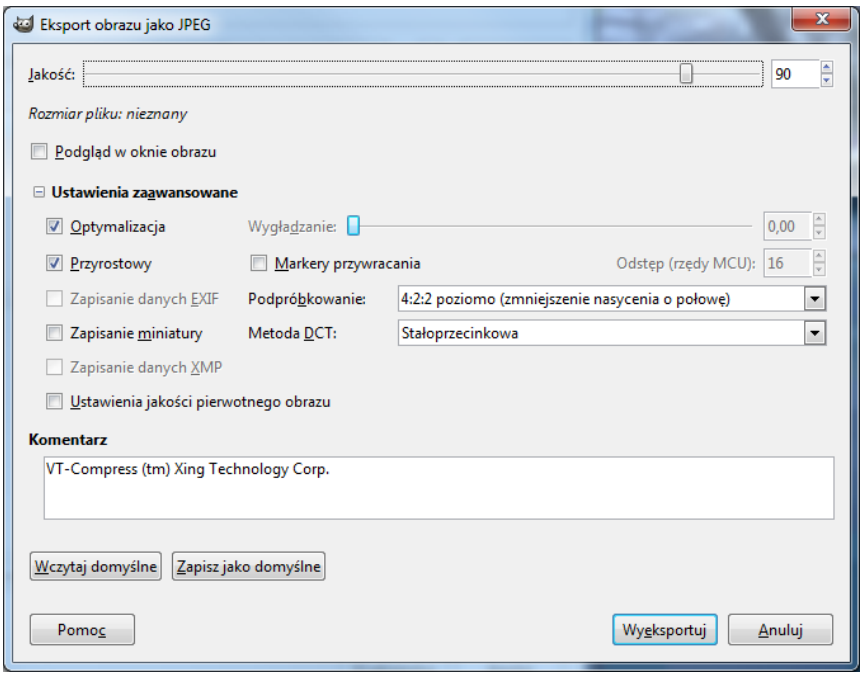
JPG quality	Compression rate	Portion of the image
100%	4,8 : 1	
90%	17,5 : 1	
80%	27,9 : 1	
70%	36,6 : 1	
60%	46,5 : 1	

50%	53,2 : 1	
40%	62,3 : 1	
30%	77 : 1	
20%	101 : 1	
10%	162,7 : 1	

Based on the analysis above:



For the analysis GIMP software was used. Settings of the export to JPG:



The lower quality of JPG compression the higher image distortion. In images compressed with 100% or 90% quality it is hard to see a difference between them and the original. Acceptable distortions occur also with 80% or 70% quality JPG compression. Below these values image's detail are very distorted. Therefore camera used as CAM2 can't compress with quality lower than 70%. For 100% quality compression the compression rate is 4,8:1 and with 70% it is 36,6:1. Because the difference between 100% and 90% quality is negligible camera should not compress with 100% quality. Compression quality of CAM2 should be between 70% to 90%. With this compression the size of one 0,3Mpix image is between 30kB to 80kB. Everything depends how many details are in the given image. These value should be treated like approximations.

4.5.5 SIZE OF FILM FROM CAM2 CAMERA

CAM2 camera's (below solar panels) task is recording of the opening of the deorbiting structure. Camera will take series of photos with VGA resolution (0,31Mpix) with the rate of 5 frames/s. Output data should be in JPG format. The film's duration should be 30s.

Without the compression the series of photos with frequency of 5 frames/s and the duration of 30s would have approximately 132MB (640pix * 480pix * 24bit * 5frames * 30s). Sending this film with UVF or VHF transmitter (10kbps) would take 32,5h. It isn't possible, so the images need to be compressed with JPG algorithm.

Images from CAM2 don't need to be high resolution and high quality. JPG 50% quality compression should be sufficient. Based on analysis in previous section we can conclude that compression rate in this case is 53,2:1. Of course it depends on the given image details. With this quality we can safely assume that minimum compression rate will be better than 15:1. With this coefficient film would have the size of 8,8MB. With transfer rate 10kbps it would take 36 communication sessions (each lasts for 15 minutes) to send it to Earth. It is very difficult to do, so the size of the film need to be further reduced.

We can lower the resolution of the film to 320x240 in the output format of YUV camera - luminance, chrominance, chrominance in 4:2:2 proportions. With these parameters uncompressed film would have the size of 11MB (320pix * 240pix * 8-bit * 5klatek * 30s). With the lower quality and JPG compression rate 15:1 the size of one film would be less than 1MB. For 10kbps transmission rate one communication session would be enough to send this film to Earth.

4.5.6 SELECTION OF MEMORY CAPACITY FOR IMAGES, FILM AND PAYLOAD DATA

NOR Flash memory holds images, films and payload data. On the order they can be sent to the ground station. Photos will be saved in compressed (with JPG algorithm) form and the film would be saved as series of JPG photos with lower quality.

NOR Flash memory needs to hold at least 20 compressed photos (1Mpix) 200kB each, 10 compressed films (320x240, YUV 4:2:2, 5klatek/s) 1MB each and 10MB payload data. Calculations:

20 photos * 200kB =about 4MB

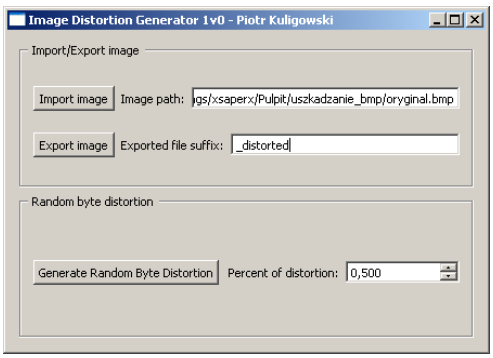
10 films * 1MB = 10MB

The minimal capacity of NOR Flash memory is 24MB (4MB + 10MB + 10MB). The closest standard capacity is 32MB. NOR Flash memory with this capacity is available.

4.5.7 RANDOM DAMAGING OF MEMORY CELLS

Ionizing radiation, after sufficient time causes random damage to memory cells. It is happening with SRAM, NOR Flash and NAND Flash memory alike. Good example of this phenomena are memory and systems glitches in the Galileo satellite, which explores Jupiter. Ground control modified its software so it could bypass damaged memory cells. Galileo in this time took four times that much radiation than it was designed to.

Program simulating random memory cells distortion



For the purpose of testing the impact of the memory cells distortion to the integrity of the image data and telemetry data the team created the program. Its task is to damage semi random places of input file in the semi random way.

By clicking "Import image" we are loading selected image. Next we need to enter "Percent of distortion" and then click "Generate Random Byte Distortion", after that we click "Export image". Deteriorated image is saved in the catalog as the original one but with added suffix. In the future program will be updated with additional simulation options to deteriorate memory cells.

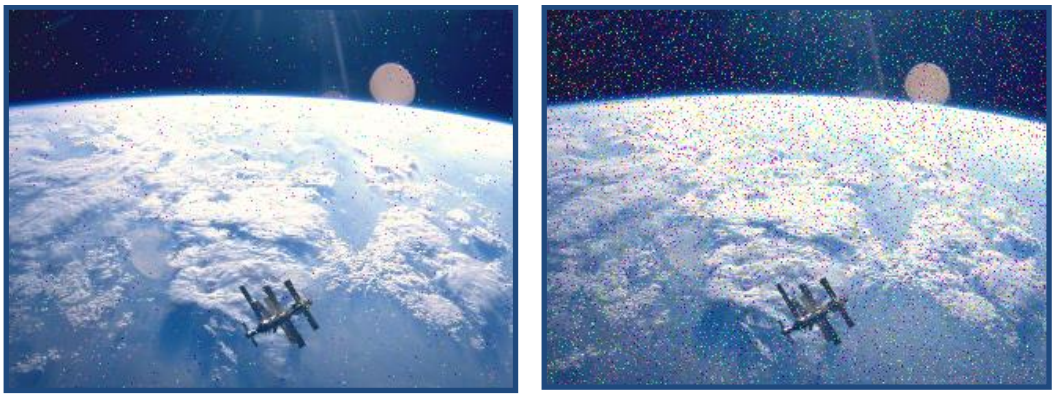
Original image – without deterioration

To examine the effect of random deterioration of memory cells on the integrity of the image data in different formats, we will use the following image (320x240 pix):

Effects of memory deterioration of BMP image

An uncompressed image saved as a bitmap BMP (Windows format). Specified number of bits corresponds to one pixel. Deterioration of a portion of bits causes a loss of some image data. Even if in such an image has 10% of its pixels damaged in random places it is possible to read it.

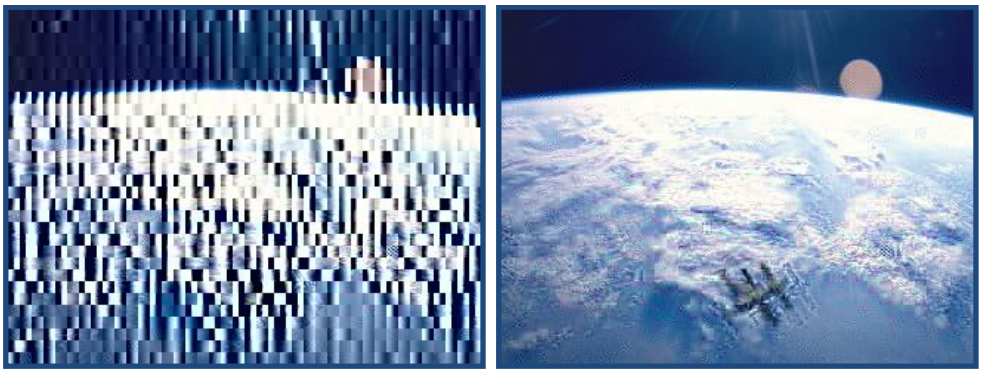
If the BNP file's header is damaged it is possible to copy it from another image with the same size and format.



In the bitmap BMP on the left side 1% of the file was damaged and on the right side the 10% was damaged. In spite these damages both are readable. In the second image BMP's file header was also damaged. It was copied from another BMP file with the same size.

Effects of memory deterioration of JPG image

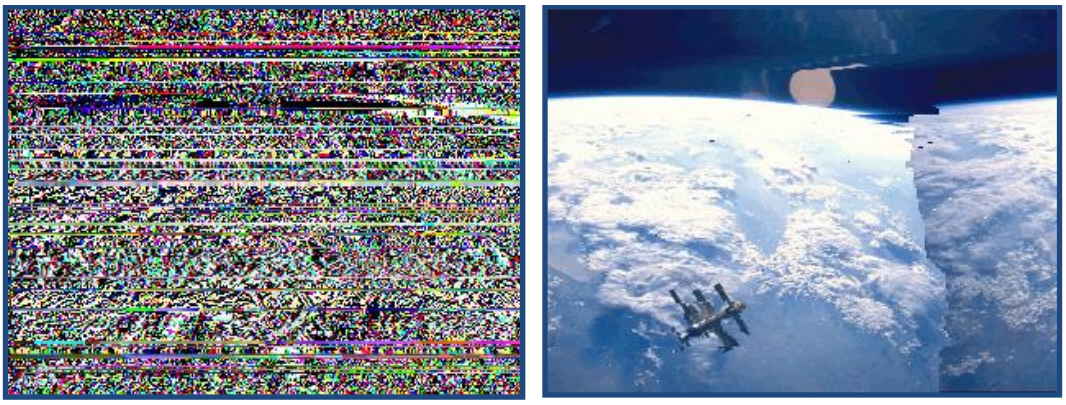
Compressed JPG image is much more vulnerable to memory cells deterioration:



On the left side a JPG image file deteriorated with coefficient of 0.1%. The picture is unreadable. On the right, the same image with the repaired JPG header. JPG file has a fixed header that was copied from another image. JPG image has become more readable. JPG images of the deteriorate coefficient above 0.1% are usually not recoverable. This example shows that the JPG files are vulnerable to both the amount of damage and at the place where the image memory has been damaged.

Effects of memory deterioration of PNG and GIF images

PNG and GIF format were also tested. The image in the first and second format are even more susceptible to damaging individual bytes. Even a single failure within the file, not the file header, makes it impossible to read.



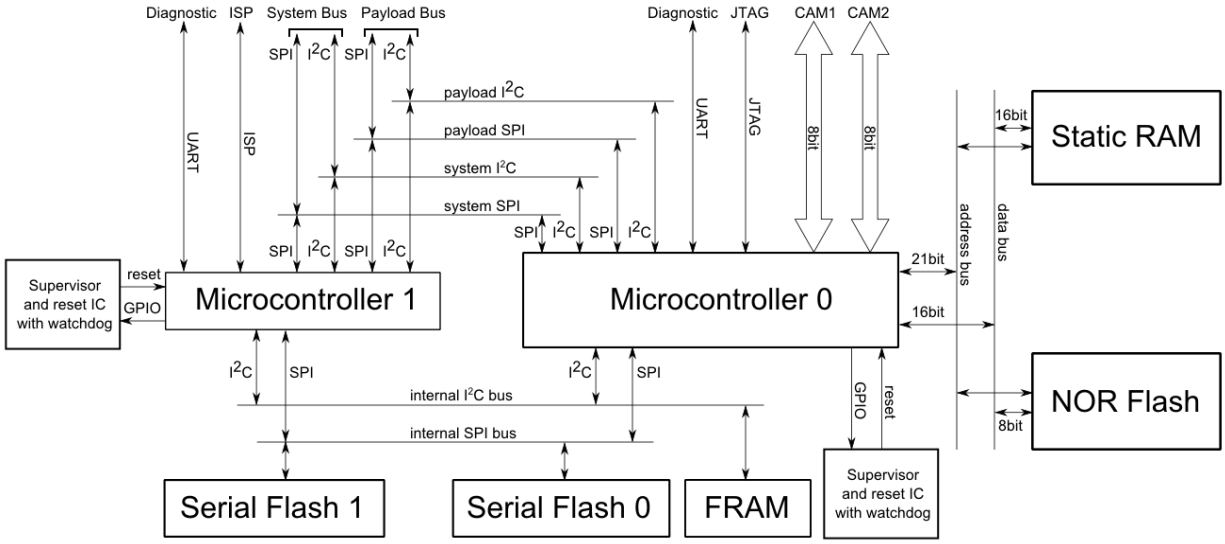
On the left side there is a PNG image and on the right side there is a GIF image. Deterioration coefficient was 0.1%. In the first picture we can see irregular noise and in the second one most of the image is unreadable. It wasn't possible to recover these images even after switching the header or switching few dozens of bytes.

Preventing the effects of image deterioration

Memory deterioration causes the loss of portion of information about the image. If there is no correction code, the situation where all received images are unreadable can occur. To prevent this in the next project's phases we need to find safer way to hold images so that single, randomly deteriorate memory cells don't cause the loss of image data.

4.7 GENERAL BLOCK DIAGRAM OF OBC ARCHITECTURE

The following block diagram shows conclusions from OBC architecture analyses in previous sections:



- **Microcontroller 0 – OBC0**

Main microcontroller which will have enough computing power to compress and send images. It will receive photos from CAM2.

- **Static RAM memory OBC0**

Pamięć do przechowywania aktualnie trwających obliczeń, np. buforowanie zdjęcia przesyłanego z kamery do pamięci Flash, przechowywanie zmiennych, itp.

- **NOR Flash data memory**

It holds photos, films and payload data collected during the mission. Works like hard disk in PC.

- **Microcontroller 1 – OBC1**

Its task is to partially substitute OBC0 in case when it malfunctions. It is much less advanced than OBC0. Its level of integration is much smaller than OBC0's, which will allow to prolong functioning even with increased ionizing radiation.

- **OBC1 Flash memory**

Serial Flash memory for holding OBC1's data. It isn't necessary for OBC01 to work but it does allow holding experiments results after OBC0's failure.

- **The arbitrage between OBC0 and OBC1**

Microcontroller 0 and 1 should be powered independently and should be turned on by the EPS. Just after OBC's start EPS questions OBC0 and if it doesn't respond EPS turns it off and turns on OBC1. Even though EPS turned off OBC0's power, OBC1 can turn OBC0 on. This was OBC0 can be reconfigured from Earth using OBC1.

- **Oversee and restart system with watchdog**

In case of the situation when OBC0 or OBC1 becomes frozen up, e.g. because of stack-overflow, watchdog system will restart it. Oversee and restart system is making sure that resetting is working correct. Cyclical restarts of OBC (only in OBC0), e.g. every few hours, can reduce the chance of stack-overflow etc.

- **Measurements data memory – Serial Flash 0**

On-board computer will measure every now and then current, voltage, temperatures etc. Serial Flash memory, connected with SPI interface, will hold these data which will be sent to Earth during the next communication session. It can replace NOR Flash memory as a payload's data memory. If OBC0 malfunctions it is disconnected from the power. Measurement's data memory needs to be power either from OBC0 or OBC1, so that we can access these data even when OBC0 malfunctions. FRAM memory's powering is handled in a similar manner.

- **Settings data – FRAM**

It holds system's configuration data and deteriorated memory cells maps. It is powered in similar way to Serial Flash 0 and it works in RAID 1 configuration.

- **Internal I2C and SPI buses**

They transfer data within the OBC subsystem.

- **CAM2 camera's interface**

8-bit parallel data buses which connect camera with OBC. It transfers image's data from camera. With payload's data bus (I2C or SPI) camera receives commands.

- **OBC0 and OBC1 diagnostics**

Two independent UART interfaces from which one connects with OBC0 and the second one connects with OBC1. With them we can connect PC to OBC and change its settings, without programming it. We can use it for tests before the launch.

- **External I2C and SPI buses**

I2C and SPI buses are lead out of OBC. Additionally they were divided into system bus and payload bus. I2C bus is used for controlling satellite's subsystems and transferring small amounts of data, and SPI bus is used for transferring large amounts of data.

5 SELECTION OF COMPONENTS FOR THE OBC0 1V0 PROTOTYPE

Based on previous analysis we can conclude that the best solution for PW-Sat2's OBC is an architecture with two different, independent microcontrollers. In previous sections diagram block was presented with the regard to memory types.

Two prototypes will be built: OBC0 (main) and OBC1 (reserve). They will be in the form of two independent PCB boards so that the programmers would have the systems for software's testing and development. Independent boards will allow to limit the costs and accelerate the work in the initial project's phase. The main objective is to create the trouble-free prototype of the main on-board computer. When OBC0 and OBC1 will be developed well enough then we will connect them into one PC-104 standard board.

Memory blocks of OBC0

On main OBC's board following memory blocks will be placed:

- SRAM – memory for holding the results of current tasks - volatile memory
- NOR Flash – memory for holding photos, films and payload data - non-volatile memory
- Serial Flash – memory for holding measurements data (power, voltage, current etc.) - non-volatile memory
- FRAM – memory for holding settings data and for information about currently processed threads

Power system for OBC0

Externally OBC will be powered with a voltage of 12V, so the prototype should have converters which will change it to 5V and 3.3V. Power system should measure the current drawn by each circuit inside the OBC, their voltage and temperature (current measurement of each of the systems will allow the creation of the most optimal power management algorithm). Power of each of the memory can be switched on / off with the transistor switch. Enabled will be only memory currently being used, which will save energy.

OBC0 communication interfaces

I2C, SPI buses and parallel interface for cameras (for prototype of CAM2 camera) will be placed externally. Architecture requires at least x3 I2C buses, but in prototype only 2 will be needed (we are skipping system and payload bus for simplicity).

5.1 MAIN MICROCONTROLLER

Requirements for main microcontroller:

1. Ability to receive data from camera and saving to Flash memory. We don't need it to compress images "on the fly" because images from camera will already be compressed.
2. Ability to change core's clock–clock rate is lowered when satellite goes into dormant state. Lower the clock rate is the lower power consumption is..
3. Hardware handling of I2C (Inter-Integrated Circuit), SPI (Serial Peripheral Interface Bus), UART (Universal Asynchronous Receiver/Transmitter), parallel memory interface, SRAM (Static random-access memory), NOR Flash parallel interface.
4. Timers set for precision timing.
5. Compatibility with various operating systems, such like eCos, FreeRTOS or embedded Linux.
6. Dormant state/energy saving state.
7. Minimum of x3 I2C, x3 SPI, x1 UART and free GPIO ports for parallel camera interface.
8. At least one successful mission on orbit.
9. DMA for fast transferring data inside OBC.
10. Operating temperature that spreads at least from -40°C to 85°C.
11. 16-bit or 32-bit architecture.
12. Thanks to internal program memory with capacity of 512kB-1MB it wouldn't require additional external Flash memory. It will limit the complexity of connections.

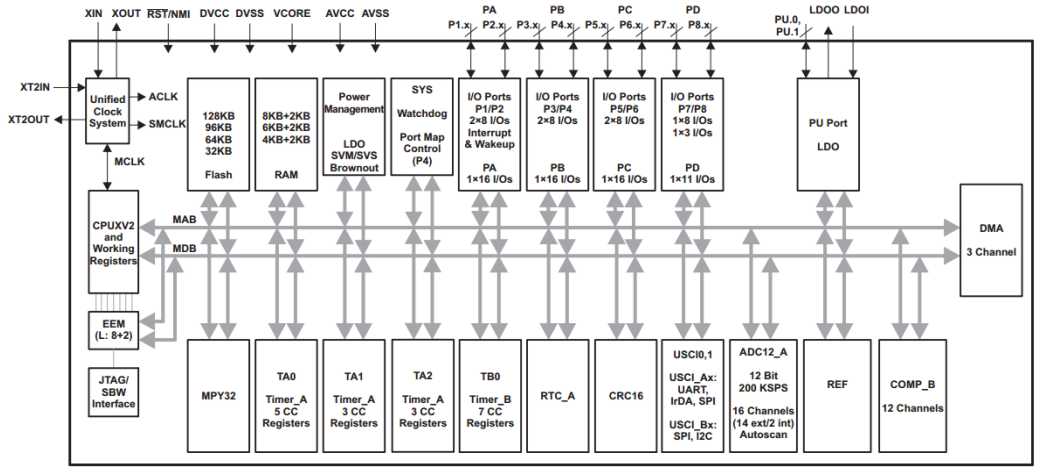
MSP430 TI

Very economical 16-bit (signal processor) microcontroller, RISC with clock rate up to 25MHz. Their characteristic feature is the architecture compatible with the Von Neuman architecture. Program and data are saved in the same memory. This way self-programming is possible. Theoretically microcontroller is capable of generating code and then performing it. This mechanism can be used for software update while satellite is on orbit.

There are versions of this microcontroller where the division between program memory and data memory can be done freely. It is possible due to the fact that FRAM memory is non-volatile,

very fast and energy-efficient at the same time. This memory capacity is usually small, up to few tens of kilobytes.

Block diagram of MSP430F5325IPN microcontroller:



Everything is connected to the same address and data bus. MSP430F5325IPN parameters:

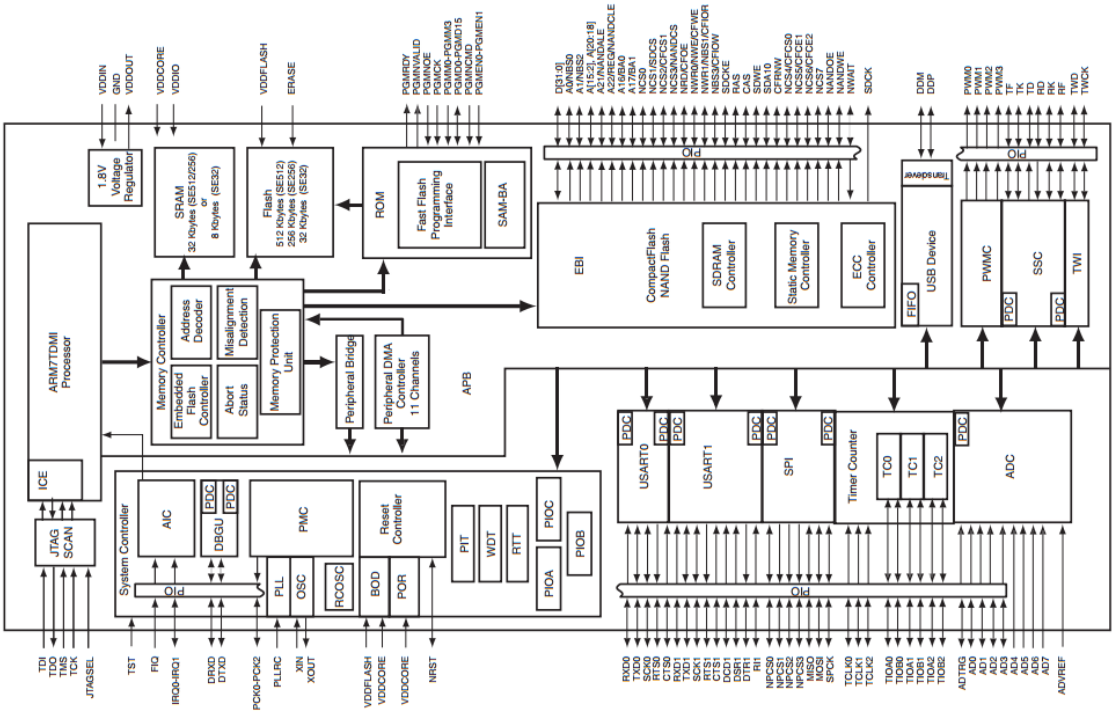
1. Voltage: 1.8 – 3.6V
2. Economical in an active state (everything turned on): 300µA/MHz with 8MHz clock rate
3. Few energy saving states: LPM3~2.1µA, LPM4~1.1µA, LPM4-5~0.18 µA
4. Activating from the dormant state in 3.5µs
5. 16-bit architecture, external memory, 25 MHz clock rate
6. Elastic system of power management
7. 4 16-bit timer
8. Two double universal and serial interfaces with I2C, SPI or UART
9. 12-bit digital analog converter, voltage comparator
10. Operating temperature range from -40 to 85°C
11. Three-channel DMA, simple RTC, Watchdog
12. Flash 128kB, SRAM 6kB (MSP430F5329 is available with SRAM 10kB).

MSP430 microcontrollers are very often used in cubesat missions. There are even commercial OBC modules with this system (eg. Pumpkin CubeSat Kit FM430).

In Delfi-C3 satellite this module was used. During the mission problems with the onboard I2C bus have occurred.

ARM7TDMI

32-bit microcontroller's family ARM7TDMI was created in 1993 and is compatible with ARMv4T Von Neuman architecture. For many years these microcontrollers were used in mobile phones. One of these microcontrollers is Atmel's SAM7's system. Following is diagram block of AT91SAM7SE512, also commonly used in cubesat missions (modules with very similar microcontrollers are available, eg. GOMSpace's NanoMind A712D):



AT91SAM7SE512 parameters:

1. 32-bit RISC architecture
2. Internal Flash memory up to 512kB
3. Up to 32kB SRAM memory
4. Memory controller with external interface (EBI)
5. Clock rate up to 55MHzand dynamic frequency scaling
6. 20-bit and 12-bit timers and PWM
7. Watchdog and RTC system
8. USB interface controller
9. Two UARTs, one I2C, one SPI

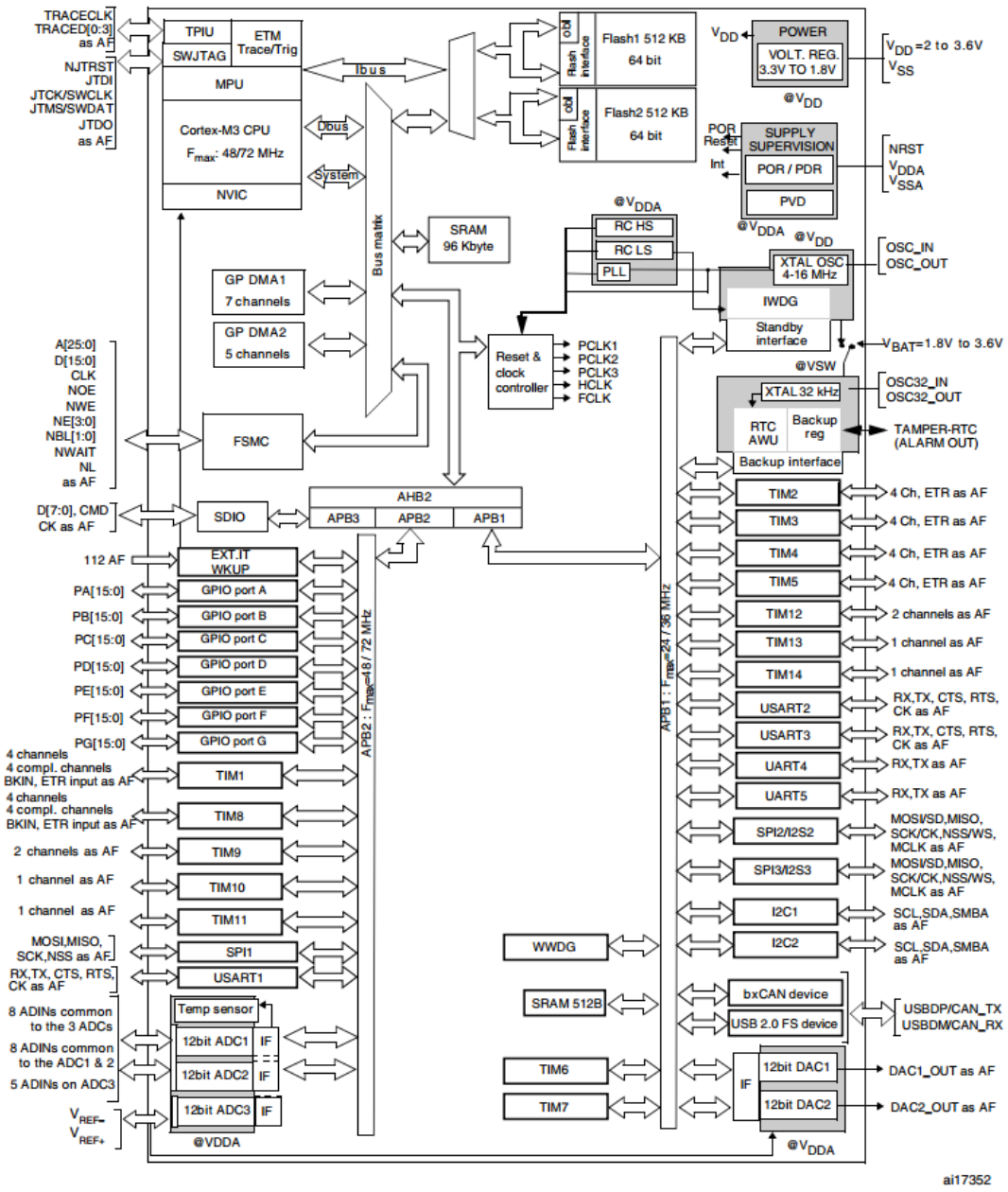
10. Eight-channel analog-to-digital 10-bit converter
11. Voltage 1.8 or 3.3V
12. In the dormant state and 25°C temperature it consumes 12-60μA, clock rate lowered to 500Hz
13. Operating temperature range: -40 to 85 °C
14. Eleven DMA channel
15. Processing orders pipelining
16. Debugging with JTAG
17. ECC error correction for Flash memory
18. 130nm process technology

ARM Cortex-M3

ARM Cortex-M3 is modern 32-bit RISC system based on the Harvard architecture. Systems with larger number of outputs have SRAM and Flash memory controllers. These systems characterized by the large number of interfaces. STM32F103ZGT6 parameters:

1. 32-bit RISC architecture
2. Internal Flash memory up to 1MB and up to 96kB SRAM
3. Advanced interrupt controller (NVIC)
4. Voltage from 2 to 3.6V
5. Power saving modes: in STANDBY with VDD=3.3C it consumes 2.5μA, and in STOP mode 50μA
6. 12-bit analog-to-digital and digital-to-analog converters
7. Up to 122 IO ports
8. Large number of communication interfaces:2x I^2C, 5x USART, 3x SPI, CAN, USB, SDIO
9. Operating temperature range from -40 to 105°C
10. 180nmprocess technology- http://zeptobars.ru/en/read/open-microchip-asic-what-inside-II-msp430-pic-z80
11. Debugging with JTAG/SWD
12. External memory's controller
13. 12 DMA channels
14. Watchdog system

A similar microcontroller was used in ESTCube-1's OBC. It functioned without a failure.



ARM Cortex-M3 jest much more advanced than any of previously analysed. They are currently the only microcontrollers with embedded systems so they are commonly known among specialists.

Selection of the main microcontroller for OBC

MSP430 systems aren't widely known and used. Generally ARM microcontrollers are used, and therefore they are better known among our programmers. In this case, where there is little time we must rely on tested and well-known solutions. Therefore we will use system with ARM core.

As the OBC's main microcontroller we will use STM32F103ZGT6 system, which is much more advanced than ARM7TDMI. This microcontroller is built from Cortex-M3 core, 1MB program's Flash memory and 96kB of SRAM memory. Due to large program's memory there is no need for usage of external memory devices for program code. System's maximum clock rate is 72MHz. Memory controller allows for connection of external NOR Flash memory (for payload's data) and SRAM memory. Furthermore STM32 was used previously in space environment (ESTCube-1) and was successful.

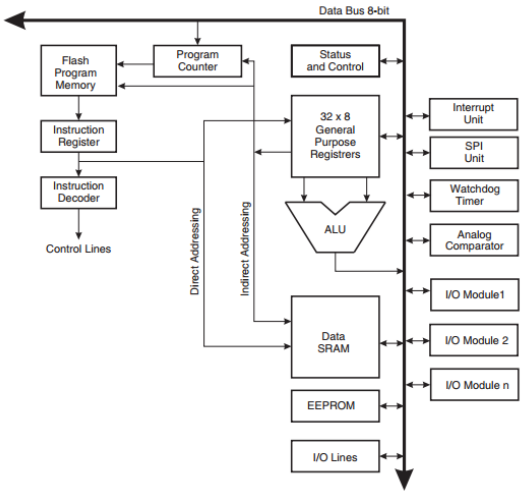
5.2 RESERVE MICROCONTROLLE

Requirements for reserve microcontroller:

1. Higher resistance to ionizing radiation than main microcontroller. Example that shows that given microcontroller can last in space environment is needed
2. The same system will be used in Sun sensor, EPS and OBC1 (reserve) so some parts of the software can be copied
3. Not complicated architecture (8 or 16-bit systems preferably)
4. Minimum of 3xI²C (control buses), 3xSPI (cameras and data) i 1xUART (debugging)
5. Dynamic frequency scaling of the core –varied depending upon the task
6. Power saving, 3.3V voltage would be ideal
7. Watchdog
8. Possibility to start real-time operating system (RTOS) with which we can copy the large portion of the program from the main microcontroller
9. Set of timers
10. Analog-digital converter (ADC)

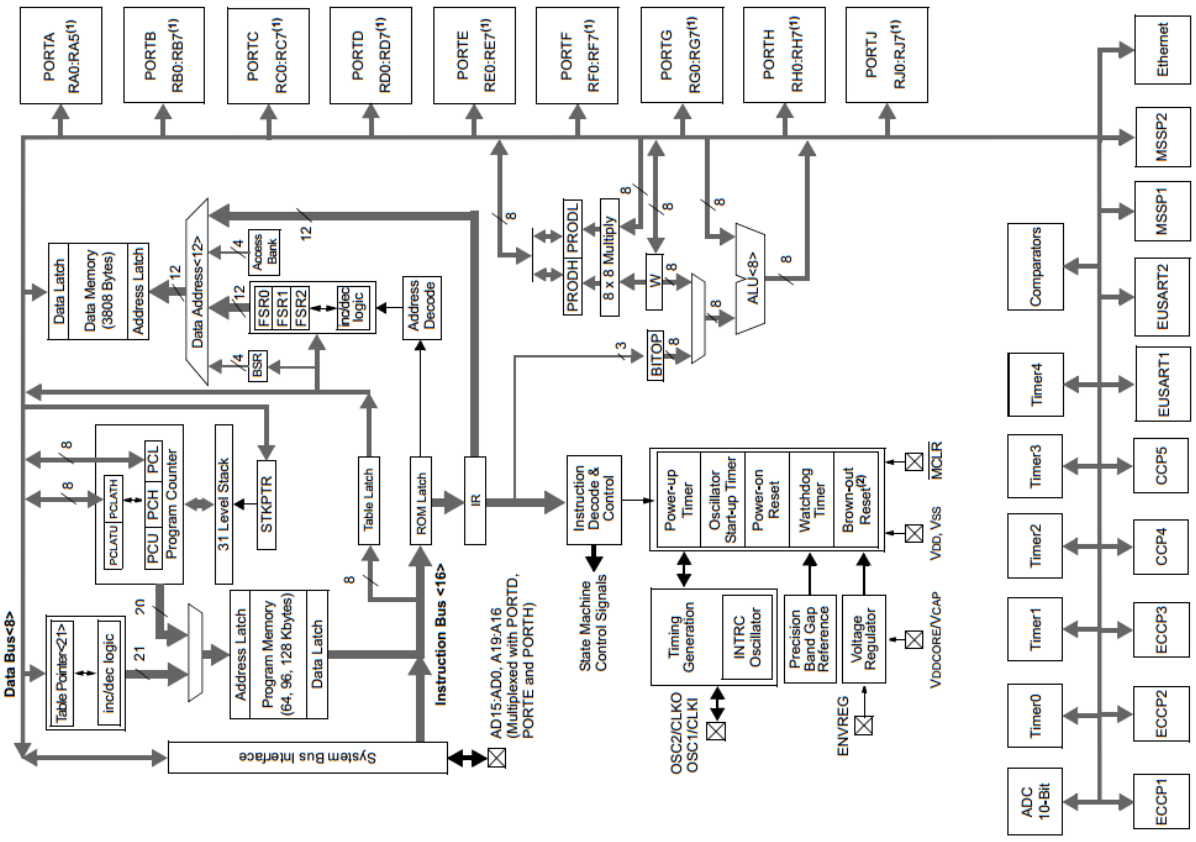
AVR ATMega128

8-bit RISC microcontroller with modified Harvard architecture from Atmel's AVR family. It has 53 I/O ports, 4kB EEPROM, 4kB SRAM, 12kB program's Flash memory, 2x USART, 1x I2C and 1x SPI. External SRAM memory can be connected. Clock rate up to 16MHz. ATMega128L can be powered with 3.3V voltage and is more energy efficient than standard version. Clock rate is set by programmer (so called "fuse-bits"). Clock rate can't be changed during microcontroller's functioning. It was used in AubieSat-1 cubesat, which functioned for over 2 years on LEO.



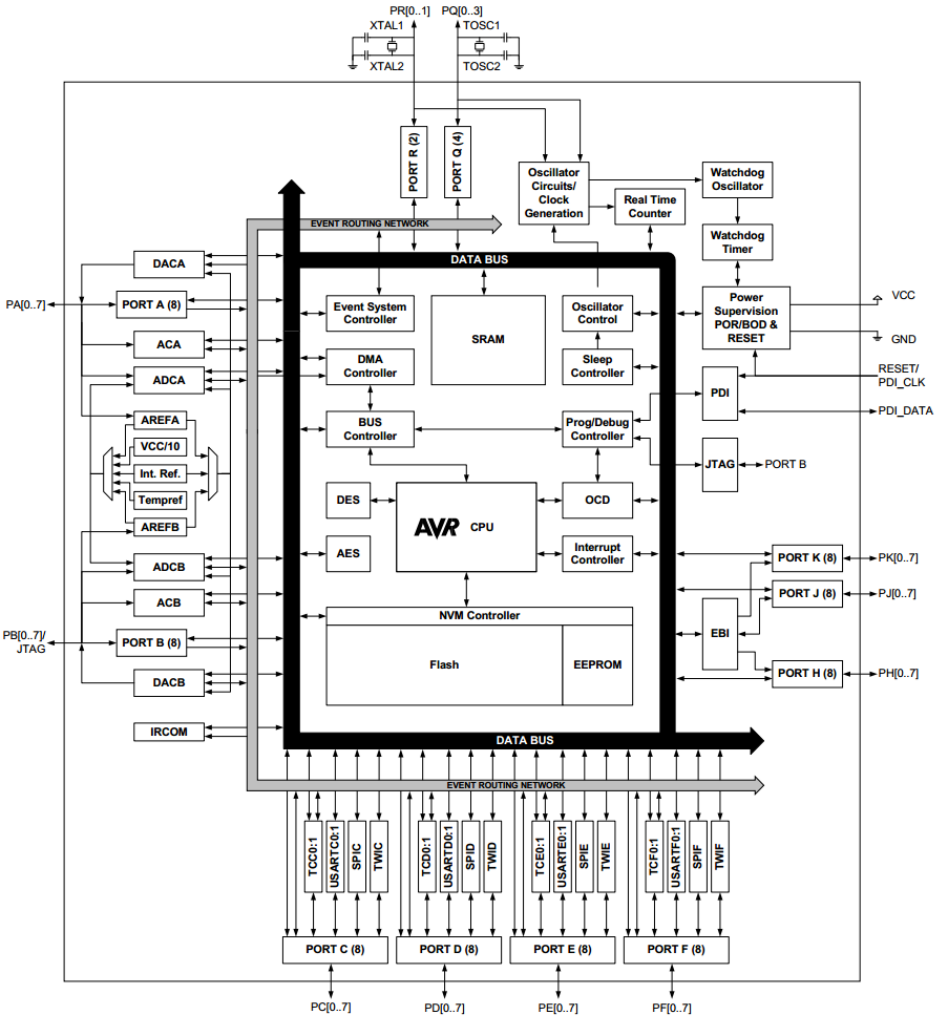
Microchip PIC18F97J60

8-bit RISC microcontroller with modified Harvard architecture. It has 70 I/O ports, 3.72kB SRAM, 128kB program's Flash memory, Ethernet, 1x SPI, 1x I2c, 2x USART and DMA. Voltage from 2.35V to 3.6V. PIC microcontrollers are commonly used in cubesat missions (eg. Tlast-1, Quakesat, AAUSat, GeneSat).



ATXMega128A1

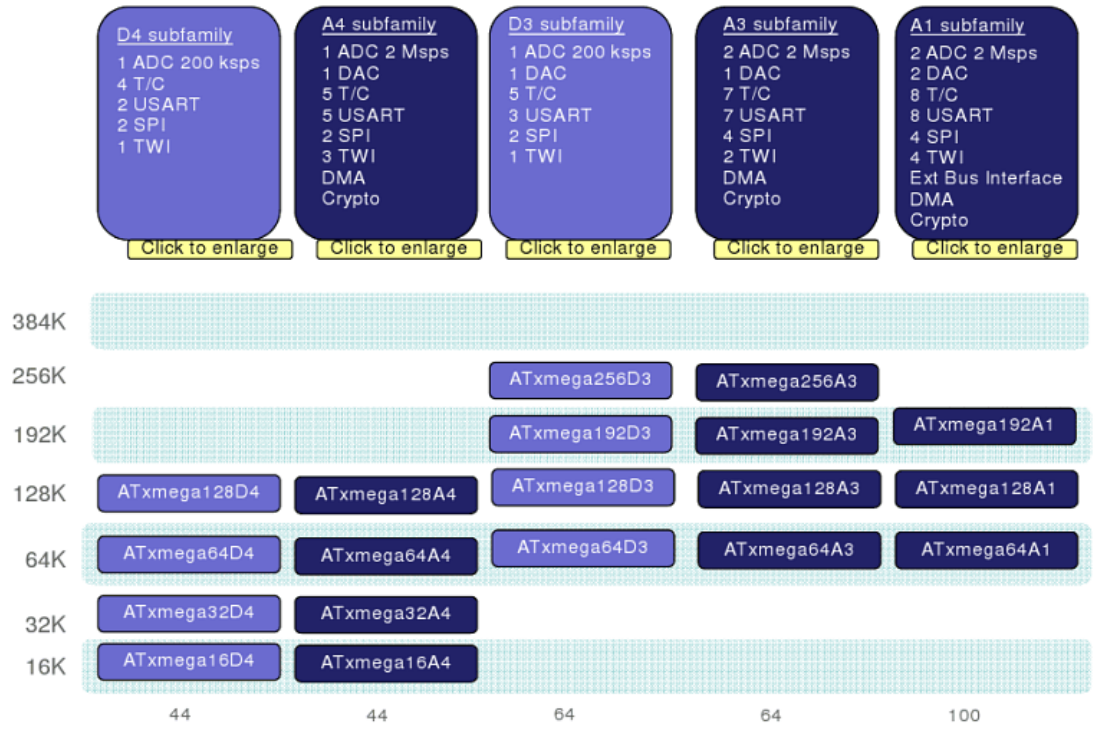
ATXMega is a modernized version of microcontrollers form AVR ATMega series (that is e.g. ATMega 128). Unlike the previous one it has more peripherals (large portion of it can be connected with ports), is more energy efficient and faster. It has 8x USART, 4xSPI and 4x I2C.



Selection of the reserve microcontroller for OBC

ATMega is outdated microcontroller, the number of peripherals is too small (I2C and SPI). ATXMega takes its place on the market. As the reserve microcontroller for OBC we will use ATXMega system.

The main criteria are: the number of peripherals, Flash and SRAM memory capacity. The following is the table with peripherals, Flash memory and the number of outputs:



On-board computer requires reserve (OBC1) microcontroller with at least 3 I2C buses (control buses), 3 SPI buses (cameras and memory) and 1 UART interface (debugging). TWI that is two-wire interface (I2C). These requirements satisfy systems from the A1 series (column to the right). On the horizontal axis number of I/O ports were put. System from A1 series has 100 of them.

As the reserve on-board computer we will choose ATXMega192A1 microcontroller. In Poland ATXMega128A1 is available so we will use it in the prototype.

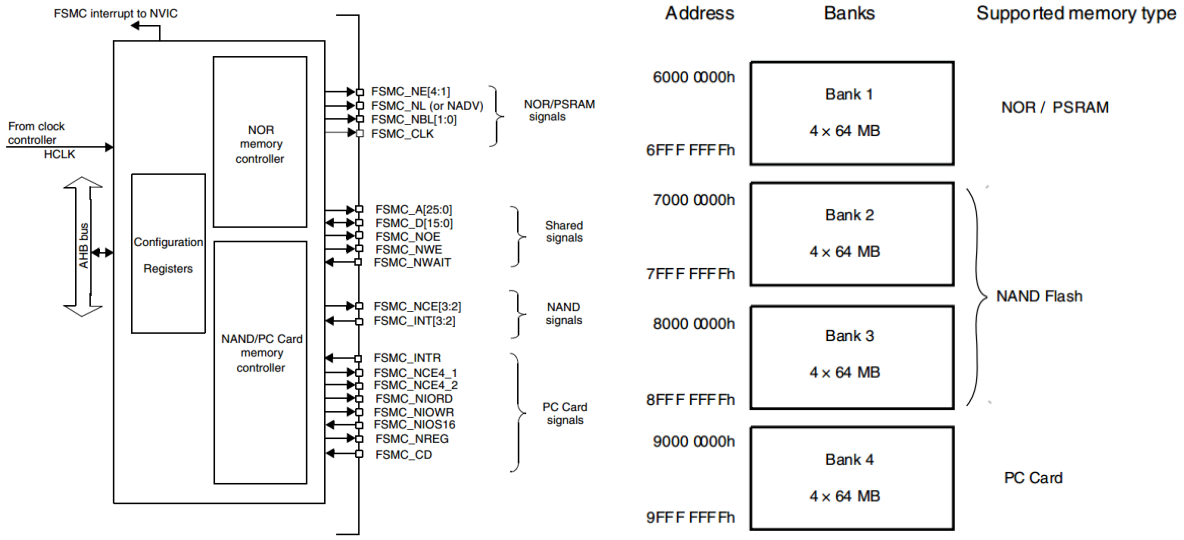
5.3 SRAM MEMORY SYSTEM

The prototype will have CAM2 camera's (640x480 resolution, VGA) interface with uncompressed output data put into it. From the analysis we can conclude that for camera like this SRAM memory of capacity of at least 1MB is required. Microcontroller STM32F103ZGT6 has memory controller (FSMC) with parallel interface with 26-bit address bus and with 16 or 8-bit data bus. This interface will be used for handling the external SRAM memory and NOR Flash memory.

SRAM memory system should have parallel interface, compatible with this in STM32F103ZGT6.

Memory controller in STM32F103ZGT6

Flexible static memory controller (FSCM) can handle external SRAM systems (Static RAM), ROM, NOR Flash, NAND Flash with ECC and PSRAM. External SRAM and NOR Flash memory need to be connected in static memory asynchronous mode, where the synchronization is happening inside of microcontroller, clock isn't provided for external memory. FSMC structure and addressing:



Connection of FSMC controller to the external SRAM memory:

Table 107. Nonmultiplexed I/Os PSRAM/SRAM

FSMC signal name	I/O	Function
CLK	O	Clock (only for PSRAM synchronous burst)
A[25:0]	O	Address bus
D[15:0]	I/O	Data bidirectional bus
NE[x]	O	Chip select, x = 1..4 (called NCE by PSRAM (Cellular RAM i.e. CRAM))
NOE	O	Output enable
NWE	O	Write enable
NL(= NADV)	O	Address valid only for PSRAM input (memory signal name: NADV)
NWAIT	I	PSRAM wait input signal to the FSMC
NBL[1]	O	Upper byte enable (memory signal name: NUB)
NBL[0]	O	Lowed byte enable (memory signal name: NLB)

PSRAM memories are addressed in 16-bit words. The maximum capacity is 512 Mbit (26 address lines).

CLK isn't provided for SRAM memory. NL and NWAIT are used only in PSRAM memory (this type of storage device isn't used in PW-Sat2). To the A[25:0] lines memory's address bus is connected and to the D[15:0] data bus. By NE[x] output microcontroller chooses the memory from which it currently reads/writes to. (so called chipselect). NOE controls output's buffor in the memory. If the NOE is set on the logical zero then the internal memory's data I/O is connected to the output of the system. Otherwise outputs are set in the high impedance state (three-state logic). Setting NBL[1]/NBL[0] causes the connection to the I/O through the external memory the more significant byte and/or the less significant byte.

Connection of FSMC to the external NOR Flash memory:

Table 105. Nonmultipled I/O NOR Flash

FSMC signal name	I/O	Function
CLK	O	Clock (for synchronous burst)
A[25:0]	O	Address bus
D[15:0]	I/O	Bidirectional data bus
NE[x]	O	Chip select, x = 1..4
NOE	O	Output enable
NWE	O	Write enable
NL(=NADV)	O	Latch enable (this signal is called address valid, NADV, by some NOR Flash devices)
NWAIT	I	NOR Flash wait input signal to the FSMC

Connections of the NOR Flash and SRAM memory are very alike. In the NOR Flash there are no pins which controls the picking of the more significant byte and/or the less significant byte of the output data. In this case, just like in the SRAM memory, maximum memory capacity of one NOR Flash system is limited to 512Mbit (64MB). The one bank consists of four such systems, so one bank of memory holds up to 256MB. NOR Flash and SRAM memory can be connected only to the first bank so the total capacity of them is 256MB. It is confirmed by memory map from previous page. This introduces a certain limits: if we use 1MB SRAM memory we can't use up 255MB Flash memory but only 192 MB.

A detailed description of FSMC system is available in the "RM0008 Reference Manual - STM32F101xx, STM32F102xx, STM32F103xx, STM32F105xx and STM32F107xx advanced ARM-based 32-bit MCUs" STMicroelectronics's document, from where above pictures are (from page 487). Memory's $\overline{CE}$ connected with NOE[x], it is a chip-select. Memory's $\overline{CE}$ connected with NOE, memory's $\overline{WE}$ with NWE, $\overline{UB}$ and memory's $\overline{LB}$ connected with FSMC's NBL[1] and NBL[0].

Selection of the SRAM memory system

In the OBC0's prototype (main computer) in the version 1v0 we will use SRAM memory with capacity of 1MB. SRAM memory system needs to be compatible with FSM memory controller in the STM32F103ZGT6 microcontroller.

Information about selected memory's systems (all of them have voltage of 3.3V):

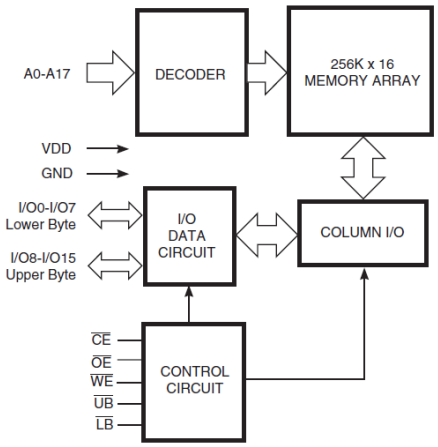
#	System	Capacity and organization	Number of systems used	Access time to data	Power in active state	Power in dormant state	Operating temperature range	Availability
1	CY7C1041DV33-10ZSXI	256kb x 16	2	10ns	300mW	30mW	-40 +85°C	Farnell.com
2	AS7C34096A-10TCN	512kb x 8	2	10ns	650mW	28,8mW	-40 +85°C	TME.eu Farnell.com
3	IS61WV25616BLL-10TLI	256kb x 16	2	10ns	85mW	7mW	-40 +85°C	TME.eu

Three SRAM memory systems with similar parameters from three different companies were chosen: Cypress, Alliance Memory and ISSI. The most energy efficient is the ISSI's memory IS61WV25616BLL-10TLI. It is the memory with 16-bit data bus, capacity of 512kB and access time of 8,10 or 20 ns. Memory with the same outputs distribution is available with the capacity of 2MB. Therefore without large software and connections changes we can update SRAM memory from 1MB to 4MB (in the previous sections of this documentation the requirements for camera's memory were described).

In Poland IS61WV25616BLL-10TLI is available on TME.eu, so this system will be chosen for the prototype. If this system will satisfy our requirements in the next version we will use this type of memory with larger capacity (available at Farnell.com).

Connecting SRAM memory to the microcontroller

IS61WV25616BLL-10TLI outputs:



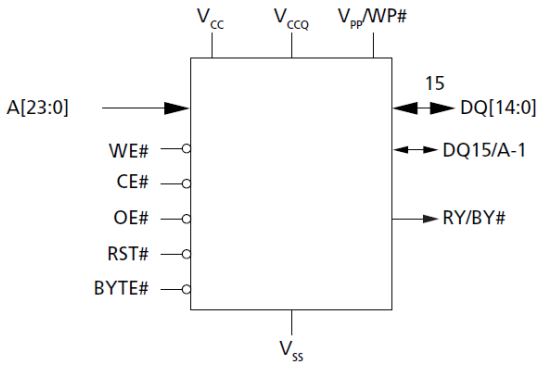
Address lines A0-A17 are connected with microcontroller's address lines A[25:0]. Memory has only 18 address lines, microcontroller has 26, and therefore 8 of them need to be disconnected. Selected memory system has 16-bit data bus I/O0-I/O7 and I/O8-I/O15, which will be connected to the D[15:0]. Memory's $\overline{CE}$ will be connected with NOE[x], that is chipset. Memory's $\overline{OE}$ with NOE,$\overline{WE}$ with NWE and $\overline{UB}$ and $\overline{LB}$ with NBL[1] and NBL[0] of FSCM memory controller.

5.5 NOR FLASH MEMORY SYSTEM

NOR Flash memory will hold images, film and payload data. The minimum capacity of this memory (taking into account previous requirements) is 32MB. Memory system with the capacity of 32MB will be selected (in the next versions we will use systems with larger capacity).

#	System	Capacity	Random access time	Process technology	Power during writing	Power in dormant state	Operating temperature range	Availability
1	M29W256GL7AN6E	32MB	70ns	65nm	60mW	0,33mW	-40 +85°C	KAMAMI.pl
2	S29AL032D70TFI030	32MB	70ns	200nm	60mW	0,00017mW	-40 +85°C	Farnell.com

In the first prototype we will use M29W256GL7AN6E memory (because of its availability in Poland). S29AL032D70TFI030 system which is more energy efficient and is made in 200 nm process technology will be used in the next versions. Such a technological process would allow for longer functioning time of the memory in an increased ionizing radiation environment (memory cells are larger and less sensitive to radiation). $\overline{RST}$ is a chip reset, RY/$\overline{BY}$ busy information, $\overline{BYTE}$ set 8 or set 16-bit mode.



NOR Flash memory doesn't have such outputs such as UB and LB. Other outputs have the same functions as in SRAM memory. Additional memory outputs are, e.g. $\overline{RST}$reset system, $\overline{RY}$/$\overline{BY}$ occupied memory information and $\overline{BYTE}$ for 8 or 16-bit mode.