

COMMAND AND DATA HANDLING SYSTEM DESIGN FOR THE COMPASS-1 PICOSATELLITE

Artur Scholz

University of Applied Sciences Aachen, Germany

arturscholz@gmx.de

ABSTRACT

This paper describes the author's approach taken in the design of a Command and Data Handling System (CDHS) for the Compass-1 picosatellite. It outlines the hardware and software layout design and the applied considerations that evolve from the CubeSat specification and the requirements for the harsh space environment. The activities for the development of the board and the programming of the code are described together with their final implementation. An outlook on the further testing and the necessary modifications as well as a closing summary is given at the end.

INTRODUCTION

Compass-1 is the name of the first picosatellite being developed at the University of Applied Sciences Aachen, Germany [1]. Since the project's initiation in September 2003 it is being managed and carried out by students of different engineering departments, with a majority being undergraduate students. Currently the team consists of ten students but the task's challenging and interesting nature attracts more students to join. The project focuses on a number of goals. Mainly the students will gain essential practical experience in realizing a research and development project from start to end. Moreover, an adequate infrastructure shall be created that enables more space engineering activities to take place at our university in the future. And definitively not least, a fully functional picosatellite is going to be built and finally launched into orbit!

The satellite is being built according to the CubeSat specification documents [2] published by Stanford and Calpoly University, which define a cubical structure with 10cm edges and a mass of not more than 1kg. Powered by solar cells, such a satellite will have an average of $1.5W_e$ for operation. Attempting to develop a spacecraft within the stringent constraints mentioned above becomes reasonable when considering the satellite being stored inside a container (P-POD) for simultaneous launch with other CubeSats, which in turn helps decreasing launch costs significantly.

The launch date of Compass-1 is not yet determined. Nevertheless it is planned to conclude the development and have the spacecraft ready for launch acceptance testing by May 2005.

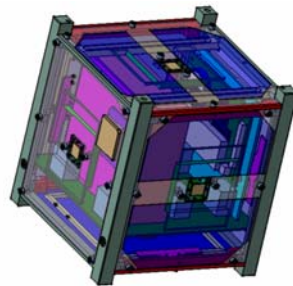


Fig 1: Compass-1 CAD model

OVERVIEW

System Overview

The CDHS has interfaces with all other subsystems. Most of them are electrical and software. There are also mechanical interfaces to the structure (to fix the board inside the cube) and to the other subsystem boards, which are mounted perpendicular to the CDHS board, also referred to as main board (as can be seen in figure 3). The electrical interfaces with the ADCS, the EPS/TCS and the COM subsystem are depicted in figure 2. Since each of the connector provides nine pins for electrical connection, the connectors for the subsystem boards provide the mechanical as well as the electrical interface at the same time.

The interface with the camera payload is being realized through a 20pin FCC cable.

The number of interfaces was ideally kept to a minimum. More interfaces increase the complexity of the system which would in turn increase the number of possible failure sources.

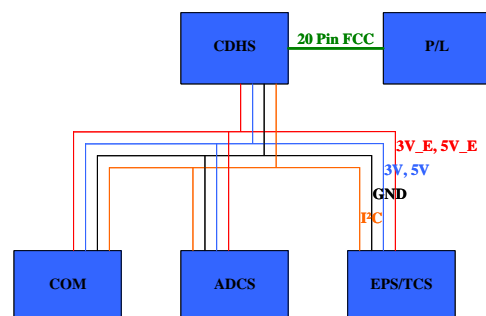


Fig 2: System-level interface diagram for CDHS

Two voltage levels are available, i.e. 3V and 5V. Both will be disconnected from power supply when the Electrical Power System (EPS) switches into power save mode. The 3V_E and 5V_E lines are permanent connected to power, which is useful for the beacon generator and other critical items.

Overview of CDHS

The CDHS is basically composed of three units, which are the micro-control-unit (MCU) and its periphery, the memory unit and the

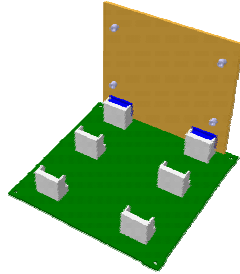


Fig 3: System integration

payload interface unit.

As the block diagram

in figure 4 illustrates, those units are all internally interfaced with each other. The communication with the other systems is done by the MCU through the I²C system bus. Each of the units composes a several components. In the case of the payload interface the number of implemented devices has become quite high due to its complex handling.

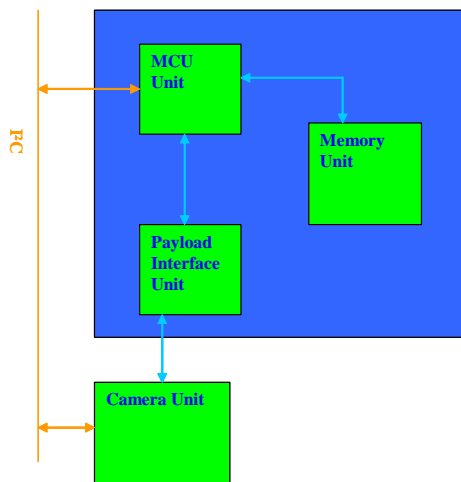


Fig 4: Top-level interface diagram for CDHS

The tasks of the CDHS are as follows:

- control payload (camera) for image capturing;
- periodically gather sensor data from other systems;
- store data (images and sensor data) in memory unit;
- receive and send command/data to COM system;
- and control ADCS (on/off).

Design Tools

For the hardware design the Protel 2004 package from Altium was used. The software provides the necessary tools for the creation of the schematic and the transition into the PCB layout. It also has an autorouter and other useful features.

The programming was done within the Integrated Design Environment (IDE), which is supplied with the target board product for the Silicon Laboratories MCU programming and debugging. It has a fully functional Assembler and a Keil 'C' Compiler with a code limit of 4kbyte.

Design Considerations

Designing a computer system for a satellite differs significantly from other (terrestrial) computers, because it has to operate in the very harsh environment of space. It is therefore necessary to pay attention to a range of constraints and requirements explained in the following.

Low cost

This is not a designated requirement of space but is generally applicable to all (satellite) projects. In the case of a CubeSat this can be understood as keeping the costs on a lowest level, due to the very limited financial budget of a university. The exclusive use of commercial off-the-shelf (COTS) products is the best approach to do so. Yet the introduced risk by using those not space-qualified devices should be minimized via extensive testing or heritage from other missions.

Small size

All parts of a CubeSat need to be very small in order to fit in the 1 liter volume altogether. Miniaturized components and SMD footprints are a good solution.

Low power consumption

The available power in a CubeSat is only a fraction of what is used by a typical room light. Hence the use of lowest power ICs with stand-by option and preferably 3V operation is encouraged.

Temperature range

The extreme temperature range of the space environment is a very critical aspect for the satellite. The thermal analysis has predicted that inside the spacecraft the parts will experience a variation of about -20 to +20° Celsius. It might even be worse. Therefore it was decided to use exclusively devices that are designed for the industrial temperature range (-40 to +85°C).

Radiation

Perhaps the most critical impact on the design of the on-board computer system has the radiation that is present at LEO (the reference orbit for the Compass-1 satellite has an altitude of 600km with an inclination of 98°). There are two distinguishable types of effects that are caused by the radiation of trapped particles in the van-Allen belt, solar protons and cosmic rays. One is the total ionizing dose (TID), which is the cumulative ionizing damage that a device experiences over a long period of time.

Fortunately the magnetic field of the earth significantly reduces those effects compared to GEO. In addition, the short mission life-time of half a year and the fact that the modern ICs are by magnitudes resistant against the present dose allow us to ignore this damaging effects [3].

HARDWARE DEVELOPMENT

Functional Overview

Core of the Command and Data Handling System (CDHS) is an 8051-based micro-controller (C8051F123, Silicon Laboratories) with powerful features that ease its handling and at the same time respond to the system requirements. It is a low power Integrated Circuit (IC) with tiny TGFP-64 footprint. A JTAG interface allows in-system programming and debugging of the internal Flash memory of the MCU which stores the flight software.

The memory unit comprises a 16Mbyte Flash device (K9F2808UIC from Samsung) that multiplexes data and command on a single eight bit bus which significantly reduces the number of pins. The footprint is a standard TSOP-48.

The payload interface unit involves a number of devices. An oscillator amplifier together with a circuit of resistors and capacitors provides the clock signal for the camera. Two voltage regulators are used, one for the camera that needs 2V5 and the other one for the FIFO that operates at 3V3. The FIFO is used to buffer the data from the payload which is streamed out at a very high frequency. A NAND logic gate triggers the transfer of one image from the camera to the FIFO. Later on the buffer content is serially transferred to the memory unit (the Flash device) via the internal data bus for non-volatile data storage.

None of those ICs is designated space-qualified. They are all commercially available devices, whereas some have heritage from other missions. In particular the characteristic of the memory unit, when exposed to LEO environment, is of interest for future missions.

Creation of Schematic

The logical interconnections of the several ICs and passive parts are visualized in a schematic. For the CDHS board it was useful to organize it in two documents, one for the connectors and one for the other devices. Of course they both correspond to the same nets. With Protel, power nets and ground nets are by default interconnected, so there was no need to explicitly draw lines for them. This contributes greatly to the readability of the diagrams. The thicker lines in the drawing in figure

5 are the internal data bus and represent a bus of eight individual tracks. This figure gives an good overview of the number of involved parts and their connection to each other. Not shown are the subsystem board connectors, since they are on the other schematic drawing.

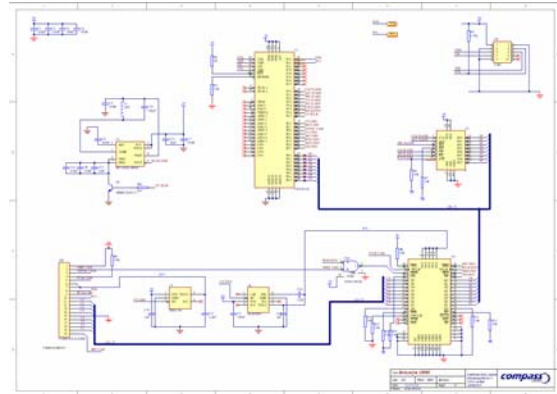


Fig 5: Schematic of CDHS device interconnections

Board Layout

The next step was to translate the logical representation of the devices and their interconnections into physical ones. Therefore the integrated library has proven quite useful as most of the footprints of the used devices were already available. Only the footprints of the subsystem board connectors and the camera connector were done manually. After placing the several parts (i.e. their footprints) on the previously defined board (it is a custom quadratic board of 95mm) the next thing to do was to lay out the tracks. First runs were carried out with the autorouter. Although the algorithm of the router is quite good, it could not satisfy with the set requirements, due to a very uncommon layout of the board. Hence the solution was to route the tracks by hand and the result is shown in figure 6 which concluded the hardware design activities so far.

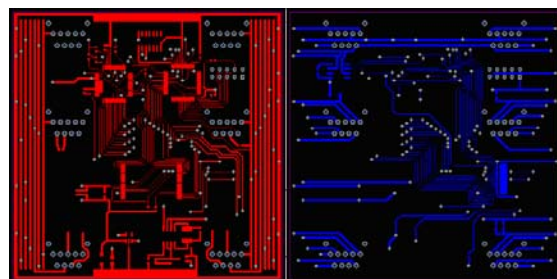


Fig 6: Front and back side of the PCB

SOFTWARE DEVELOPMENT

Although the software programming greatly builds upon the results of the hardware development process, it should not be misunderstood as being a subsequent activity. Instead, the program layout and

structure can and should be defined in parallel to the hardware.

Structured Programming

The basic key to achieve traceable and maintainable code is by following the principles of structured programming. This means that the whole program is broken into smaller parts that are much easier to handle. Those smaller parts are the modules and each of the modules can be made up by smaller modules again. By doing so, code repeating is avoided, because for each time a certain function shall be carried out, the respective module is called. The modules are grouped into different layers, according to their characteristics. For the CDHS software of Compass-1, three layers were identified as shown in figure 7. The Application Layer is the main routine, which is purely logical and totally hardware independent. The next lower level, the Device Interface Layer is the connection of the logical functions with the corresponding devices, as the name suggests. It is not hardware dependent but component independent. And finally the Low Level Drivers build the interface to the selected hardware devices. Thus the components are easily replaceable with others of the same type.

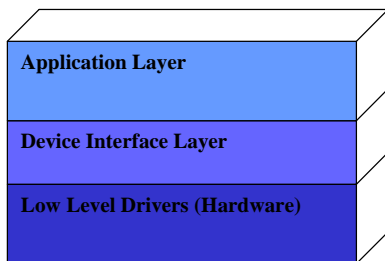


Fig 7: The layer design of the program code

The application layer was designed first by drawing a flowchart for it. Complex functions, e.g. image taking, were defined as modules with certain input and output requirements and passed on to the next lower level, where again a flowchart was created for them.

Memory Organization

The handling of the data storage orients itself on the common practice used for all types of RAM and Flash devices. The information is not sequentially addressed one byte after another but on a block and page basis respectively. This way, the allocation of a single byte can be described by using its block number, the page number and its position within a page. Figure 8 depicts this method.

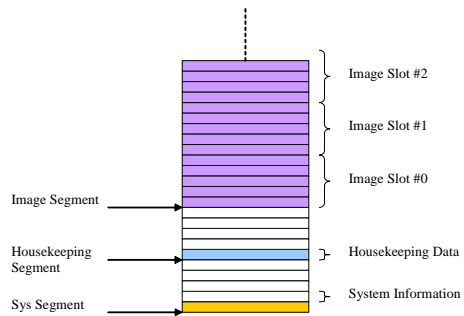


Fig 7: The layer design of the program code

Each of the horizontal bars hereby represents a block which is made up by 32 pages, with 512 bytes per page. Please note that the drawing is not to scale. In fact, the distance between the segments is 10 blocks each. One image slot reserves 20 blocks, which is large enough to store the VGA pictures from the camera.

Bus Communication

The communication, i.e. the command and data exchange, among the subsystems and the CDHS is done through the I²C bus. It is a two-wire bus system with wired-AND connection of the participants. Each member of the bus can initiate a transfer (and hence become Master for that communication) by writing a 7bit address on it, which in turn specifies and selects the device that is going to receive the data (which will consequently be the Slave for that time). Conveniently the I²C bus is already implemented in the Microcontroller hardware. Yet, the I²C standard does not have a specific protocol format and therefore a designated one was developed. Its format is given in figure 8.

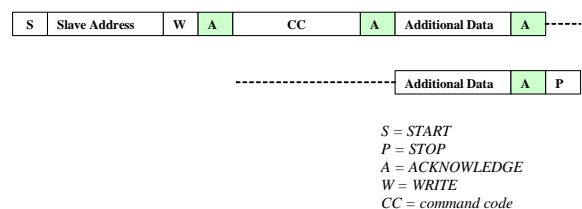


Fig 8: protocol format for Compass-1 system bus

In the Compass-1 satellite, each subsystem board has its own microcontroller that can initiate such a transfer. In order to structure the bus communication, a list of command codes (CC) was established that applies to all subsystems. A CC is 8bit long. This means that a maximum of 256 different commands can be realized.

IMPLEMENTATION

The term 'implementation' hereby refers to the manufacturing of the board and the programming of the software modules. For the programming the previously created flowcharts for each module were translated into 'C' code. Supplied with the

development kit for the C8051Fxxx MCU there was a target board that has a JTAG interface. In this way the program code could be uploaded and modified iteratively until it was finally complete debugged.

The hardware production was done in-house, where the board tracks were created by milling a copper plate. The result was quite good but still we decided to produce the board externally again to solder the devices on it.

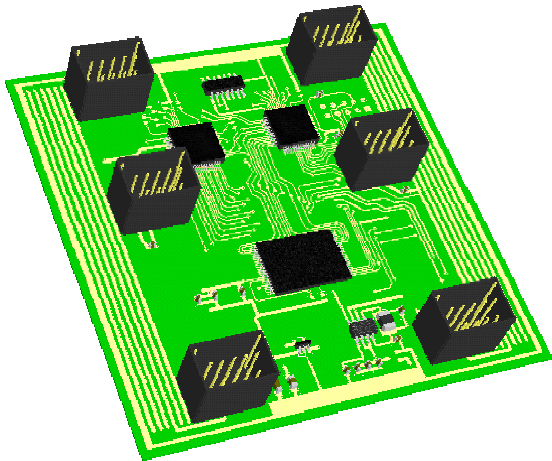


Fig 9: 3D model of the CDHS board with components

TESTING

With the completely produced board and the components soldered on it, the JTAG interface will be used for convenient testing of the program code

line by line and the debugging of it. The timings of some routines need to be adjusted as they were based on some purely theoretical numbers from datasheets beforehand. Once the system itself works, dummy boards that simulate the other subsystems can be mounted to verify a proper function when the total system is assembled.

Apart from the code testing also the hardware should be exposed to critical environmental parameters, such as extreme temperatures, mechanical shocks and radiation. This will be done according to availability and access to those facilities.

SUMMARY

It was given a brief insight into the design considerations and layout activities that were carried out for the development of the CDHS board of the Compass-1 picosatellite. So far the board is fulfilling all the requirements for an engineering model and builds the basis for the further testing.

REFERENCES

- [1] The Compass-1 Picosatellite Project at the FH Aachen. www.raumfahrt.fh-aachen.de
- [2] Calpoly and Stanford University. (2003). *CUBESAT Design Specifications Document*, Revision VIII. <http://cubesat.calpoly.edu/>
- [3] Some, R. (2003) *Radiation Models and Hardware Design*. JPL Caltech