Space engineering

Satellite HW/FW

1. Introduction

Every nanosatellite consists of several boards. Essential payloads are On-board computer (OBC) which is managing all functions on-board the satellite such as commanding, power on/off payloads or monitoring the communication lines. The next board is Radio for communication between ground station and a nanosatellite. Usually is connected to the OBC or it is part of a OBC board. The last one is Power system board which distribute power to all payloads including OBC and Radio. Also, it manages physically powering on/off other payloads. Rest of boards are usually custom payloads which are doing specific tasks.

All payloads are connected by a bus, e.g. SpaceWire, CanBus, I2C, SPI, UART or other. Due to space environment is necessary to use a robust bus, the best is in differential mode. With advantage is good to use open collector drivers to achieve more payloads connected to same bus. Here is one disadvantage, in case of failure of one payload's driver, rest of payloads will be unable to communicate.

2. Hardware for exercise

Payload programming is performed on development kits Nucleo-F446RE based on ARM microcontroller. Hardware for this exercise consists of three kits (OBC, Power board and Payload) connected by I^{2}C bus as is shown in Fig. 1. Every board has different I^{2}C address as is shown in Fig. 1. OBC has `0x01`, Power board `0x02` and payload `0x03`. Radio board is here omitted and replaced by PC connection via USB. All board are powered from USB through OBC's board.

Payload board is powered on from power board by enabling input In1 by a GPIO pin through power switch. Main I^{2}C bus is connected to the second I^{2}C bus, because payload consists Edushield which is using I^{2}C bus too. There is connected RTC clock (address `0x68`) and 4-digit 7-segment display (address `0x27`) which can be used as extension of a payload.

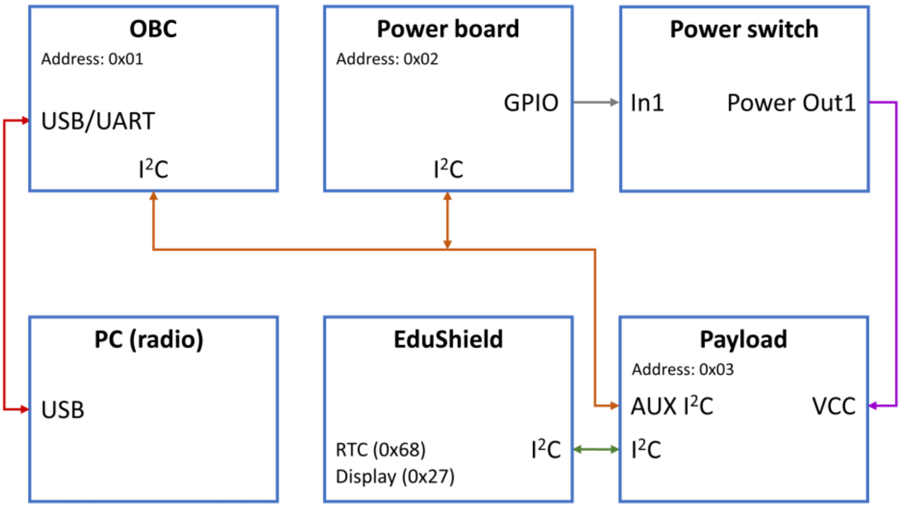

Fig. 1: Physical arrangement of payloads (OBC, Power board, Payload)

3. Programming

All boards are programmed in web-based environment Mbed.org where is created ten accounts with

name KIN**xx**
password Kin2018_**xx**

where **xx** is number in front of you (01, 02, …). Please, do **NOT** modify the password and use **your number**. Otherwise you will **edit** code of your **colleague** and vice versa.

There are prepared two basic programs, one for OBC and one for Payload. Program for Power board is not necessary to change, it is black box for you with known commands and behaviour. Communication between payloads is introduced by a header (struct) `comHeaderPacket_t` which consists of three parts:

storageID – it is identification of the command and must be unique in every payload

dataLenWrite – is length in bytes for sending data from OBC to a payload

dataLenRead – is length in bytes for receiving data from a payload to OBC

For communication between computer and OBC will be used simple serial terminal Putty, which can be downloaded from https://www.putty.org/.

a. OBC

OBC listen on UART port for commands which are then performed. List of basic commands are in Tab. 1. and in variable `cmdList[]`. Example of the command is following

```
CMD(pwrON,  "pwr on",   "Power on")
```

There are three parts, the first one is execution code (pwrON) and corresponding function must exist, the second one is the command name and the last one is description, which can be printed.

Tab. 1: Basic commands

Command	Description
help	Show all commands
pwr on/off	Power on/off Payload
pwr temp	Get Power board’s temperature
x on/off	Turn on/off payload

b. Payload

Payload could perform many tasks. In this case it is possible to measure internal (MCU’s) or external (NTC thermistor) temperature, read and/or write actual date and time, communicate with other peripherals (on edushield with dispaly), read state (simulated by a button) or drive any device (simulated by LEDs).

In the following table Tab. 2 there are inputs/ouputs of all GPIOs which can be used.

Tab. 2: GPIOs for edushield

Pin	Peripheral
D2	Button (must be pulled up)
D3	RTC square wave
D5	RGB LED - green
D6	RGB LED - blue
D9	RGB LED - red
D13	Green LED
D16	Red LED
D17	Orange LED
A0	Photoresistor
A1	Thermistor

There is also possible to read time and date from RTC chip DS1307Z via I^{2}C. Reading function is included in payload program but is not used now.

As well as Display is not implemented, but it is possible to use it. Display’s MCU listen on address 0x27 and require two bytes. The first one is position (0-3) and the second one is active segments (1 means on). In payload’s program there is prepared table for basic chars which are possible to show on display.

4. Possible tasks

1) Create a command (in OBC) and function (in Payload board) for reading temperature from internal thermometer (ADC_TEMP input) according to equation:

$$T = (3.3 \cdot V_{adc} - 0.76)/2.5 + 100$$

2) Program command for reading time from the payload by prepared function (create appropriate struct in OBC and Payload board and transfer command)

Useful links

https://www.st.com/en/evaluation-tools/nucleo-f446re.html

https://os.mbed.com/docs/latest/reference/drivers.html

https://os.mbed.com/platforms/ST-Nucleo-F446RE/