



Specifications for the Attitude Control and Determination System-XCubeSat

Dhruv Sharma, Damien Seux, Quentin Lisack, Kevin Garanger

12 April 2015

Contents

1	Introduction	4
2	Architecture of the ADCS	5
3	External interfaces	10
4	Software Architecture	11
4.1	Division of the software	11
4.2	Structure of classes	12
4.3	Definition of ports	13
4.4	List of Peripherals	13
5	Data formats	15
6	Testing	16

List of Figures

4.1	<i>Division of the software in three layers each independent of the other</i>	11
4.2	<i>Class structure for ADCS Software</i>	12

List of Tables

4.1	<i>List of peripherals with their associated ports and alternate functions (AF)</i>	14
-----	---	----

Chapter 1

Introduction

The objective of this document is to summarize in one place the specifications related to the software of the Attitude Control and Determination System, henceforth ADCS of the nano-satellite X-CubeSat developed by the students of cole Polytechnique, Paris. We have divided the document into the following sections

- Definition of the architecture of the ADCS
- External Interfaces available on the ADCS
- Sensors and peripherals on the ADCS
- Data formats used for computing
- Protocols for testing

This document shall be useful for all people currently concerned with the conception of the ADCS and eventually for students who will take over the project in the months to come. This document is however, far from being complete and additions, corrections and deletions will be made till the day of delivery.

Chapter 2

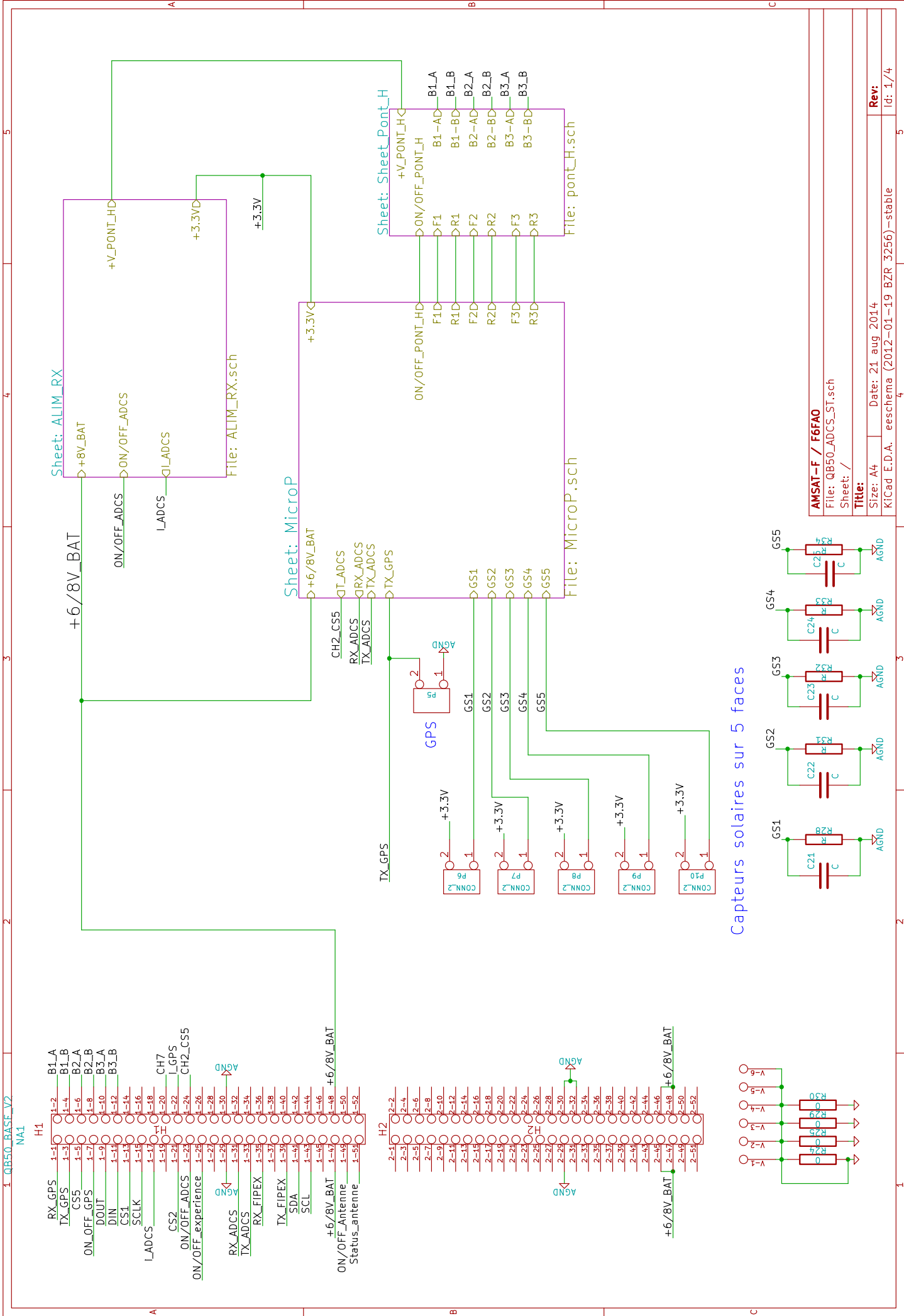
Architecture of the ADCS

The ADCS uses a microprocessor manufactured by ST Microelectronics based on the ARM Cortex-M4 Architecture. The model used is STM 32F405 RGT6

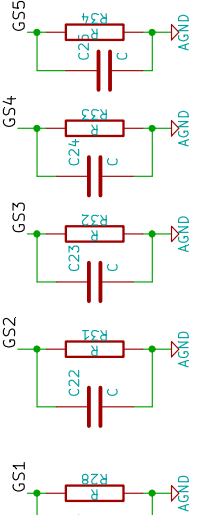
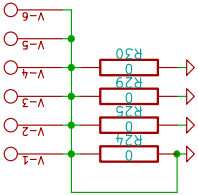
The available interfaces, sensors and other peripherals available on the ADCS are:

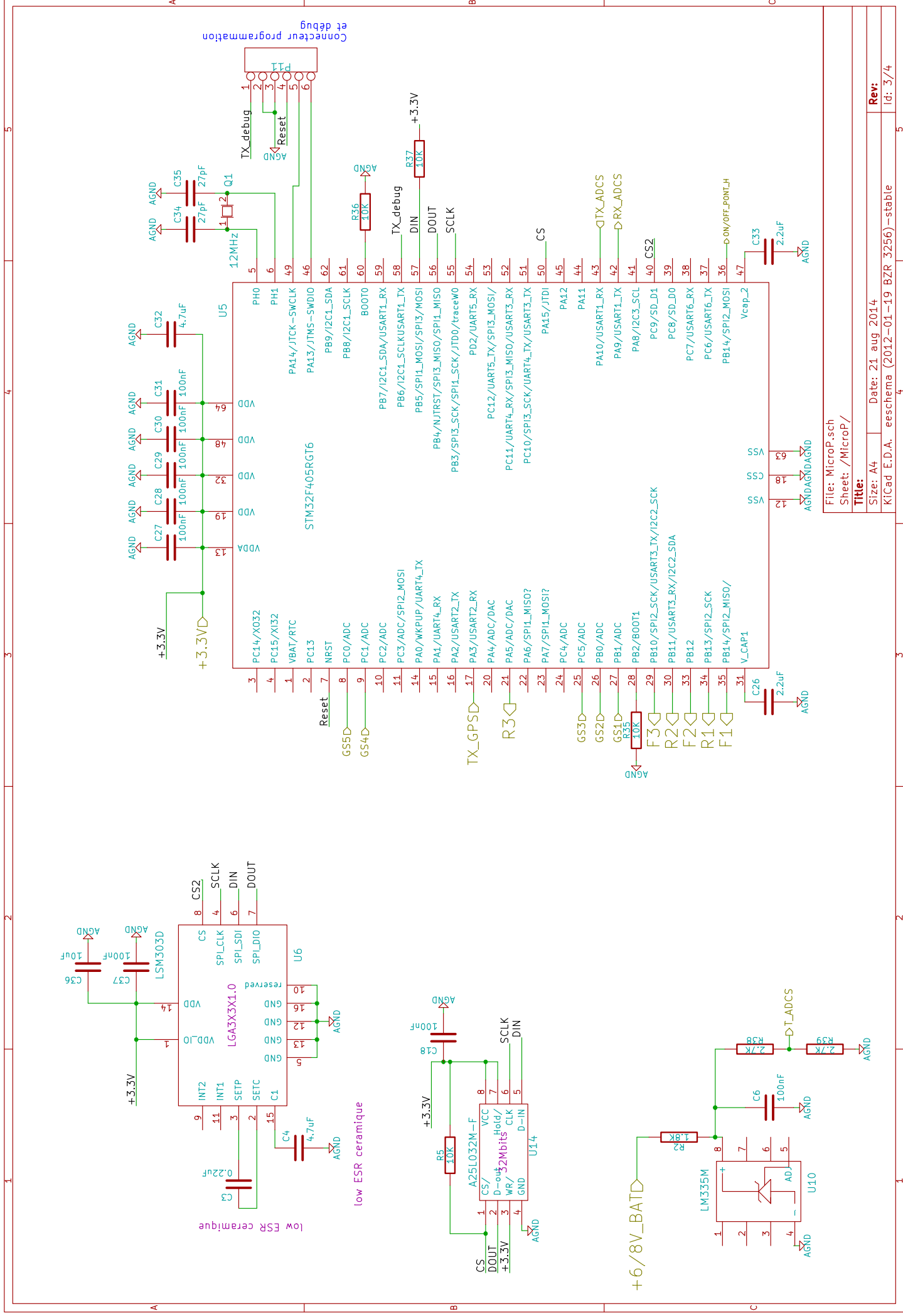
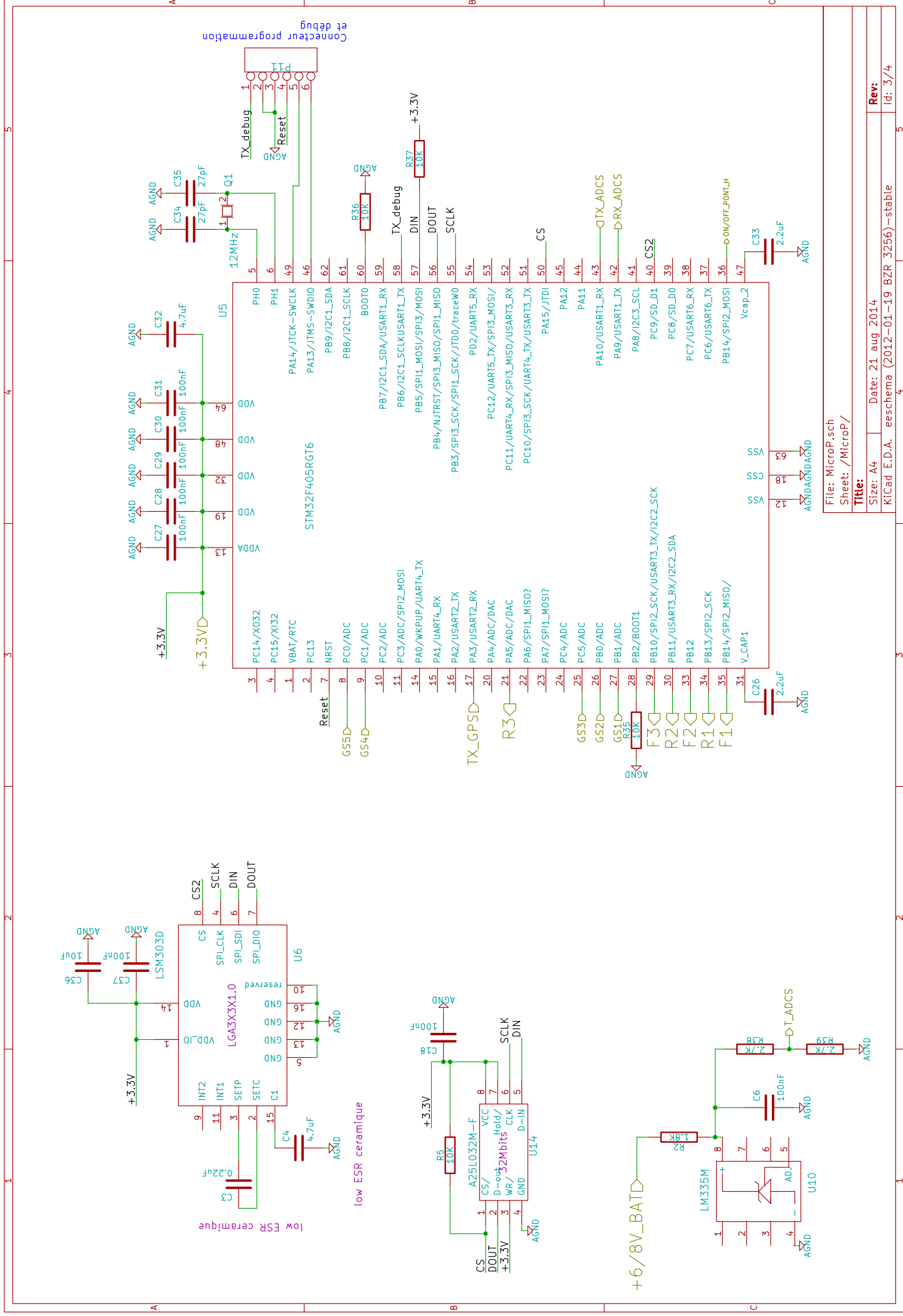
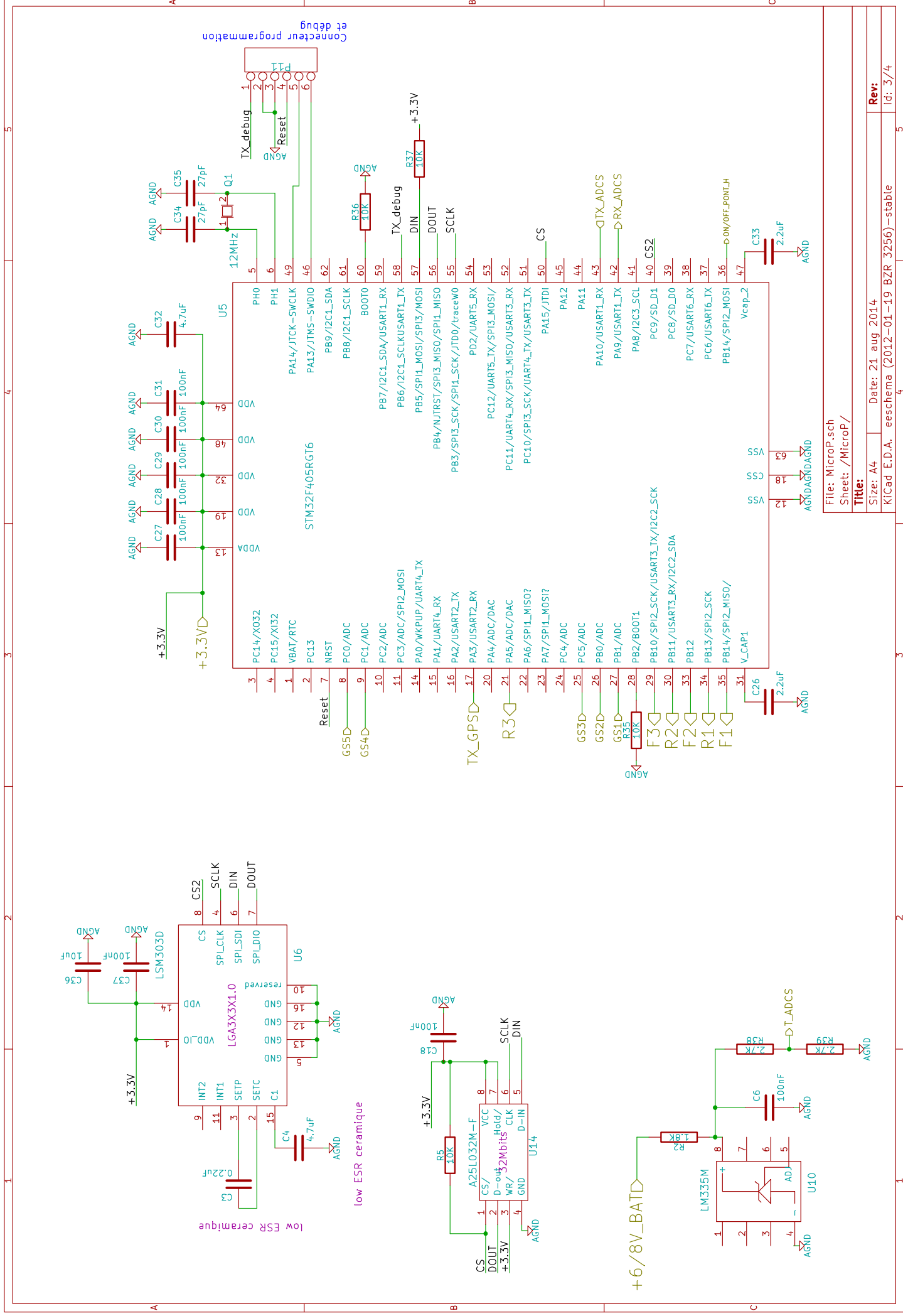
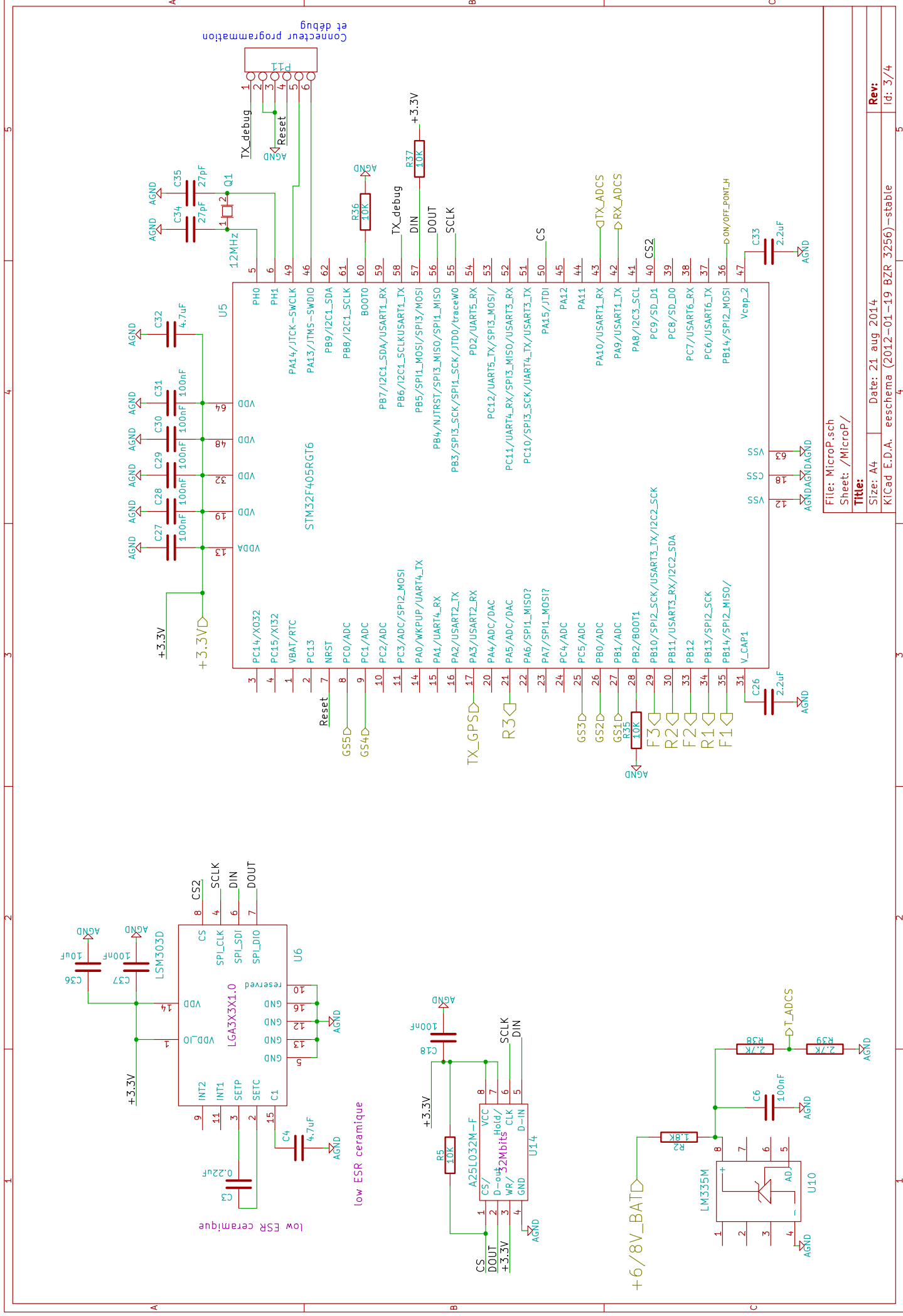
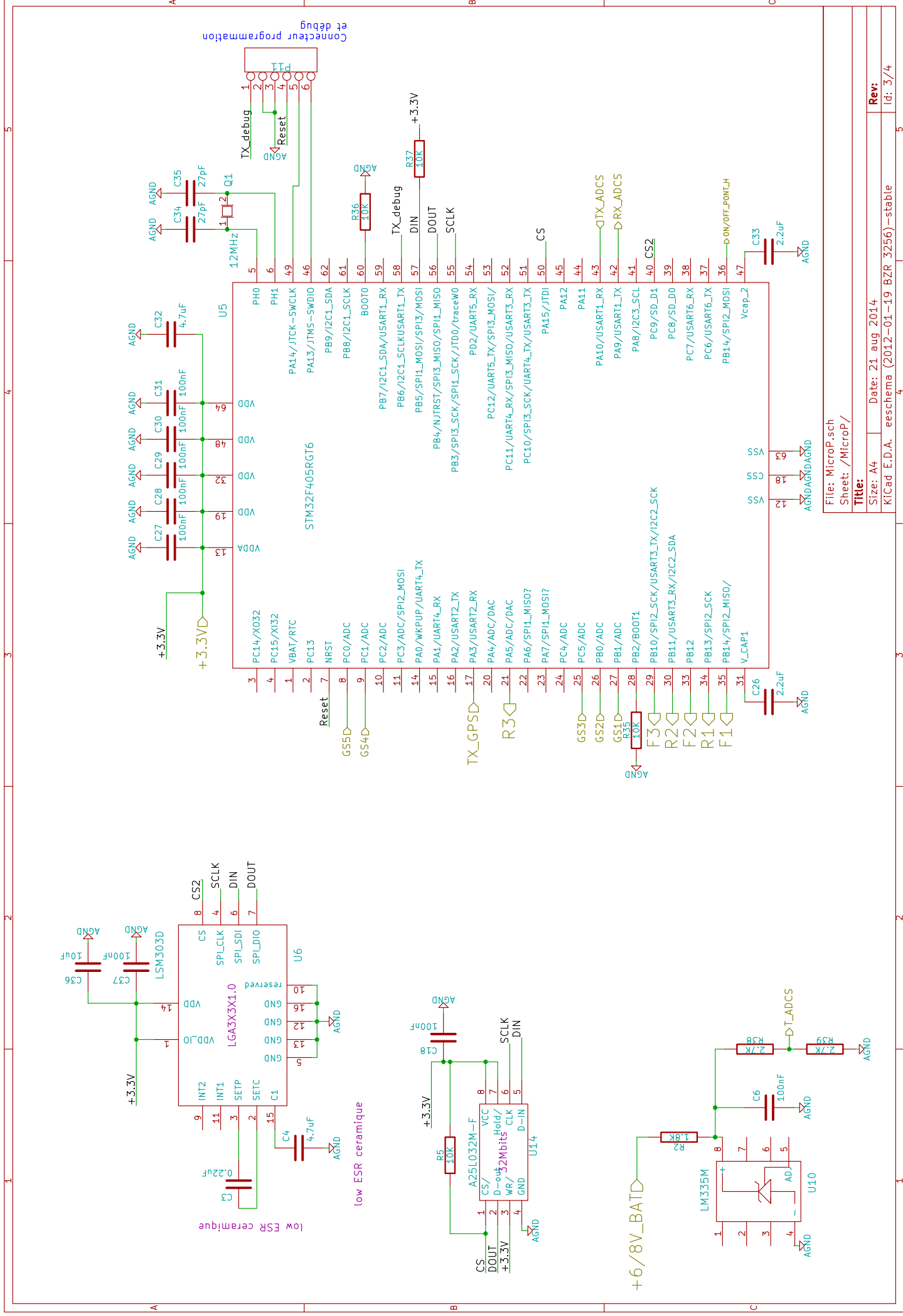
- 32 MB External Memory
- 7(or 9) sun sensors
- Magnetometer
- Gyrometer
- H-Bridges to modulate power to the magneto-torquers
- A Serial interface with the On-Board Computer (*Ordinateur de Bord* henceforth, ODB)

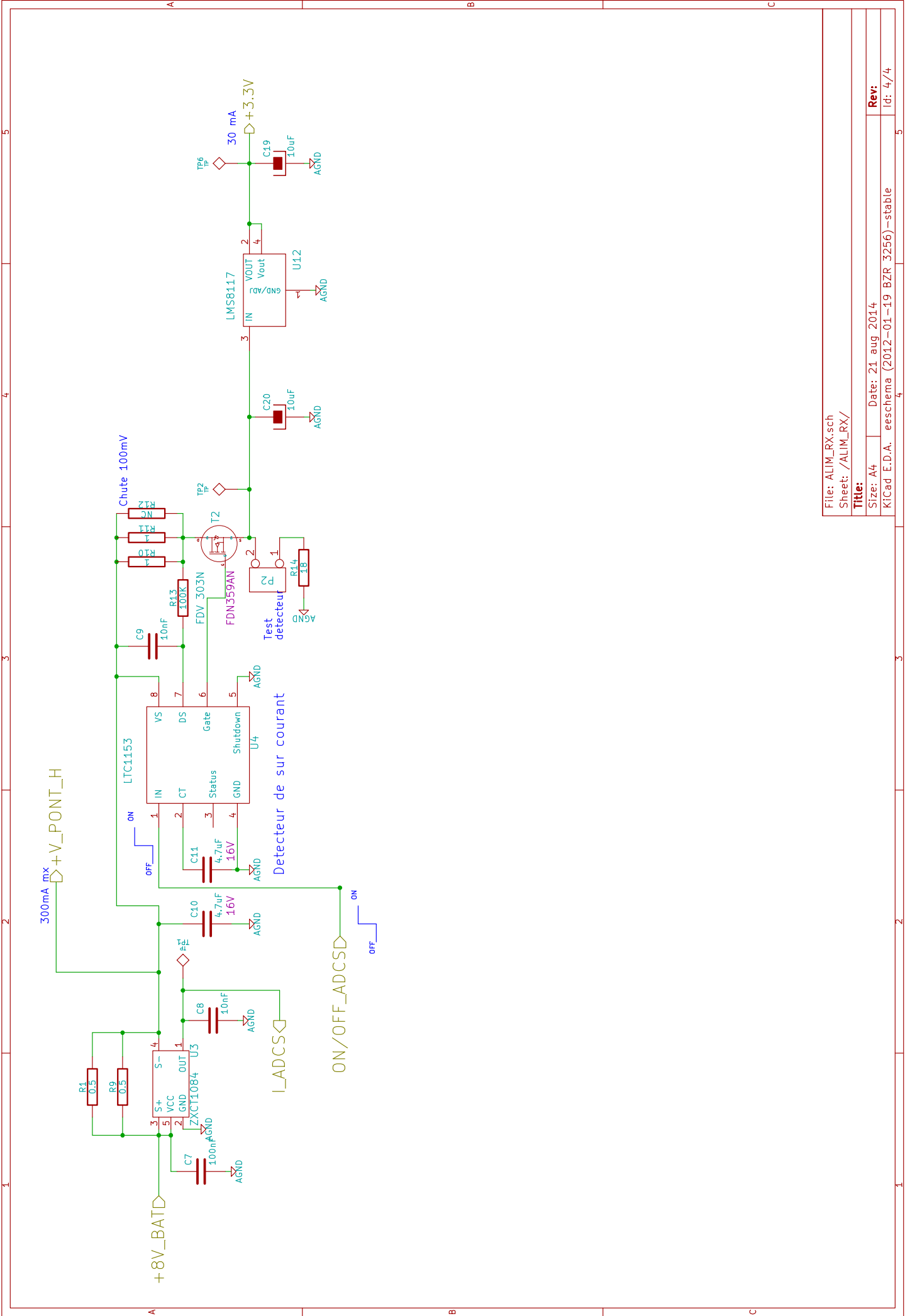
Following is the complete plan for the ADCS:



Capteurs solaires sur 5 faces







File: ALIM_RX.sch

Sheet: /ALIM_RX/

Title:

Size: A4

Date: 21 aug 2014

KiCad E.D.A. eeschema (2012-01-19 BZR 3256)-stable

Rev:

Id: 4/4

Chapter 3

External interfaces

Chapter 4

Software Architecture

This is the heart of this document. We discuss here the architecture of the software, the various peripherals and the sensors mounted on the microprocessors, the various functions associated with them detailing every time the input, output and the configuration parameters for each of the peripherals.

We first begin by detailing how the software has been divided. Next we define the various classes that we have divided the software into. While the first division is an abstract division that we made to facilitate the development of the software, the structure of classes and their interdependence correspond to how the software will actually be implemented.

Once the abstract division and the structure of classes are defined we detail the correspondence of the various ports/ pins available on the microprocessor and the peripherals connected. Next, we define the functions corresponding to each of the divisions. Irrespective of their eventual use, all functions shall find themselves in one of three categories that we shall define in the section **Division of the software**.

4.1 Division of the software

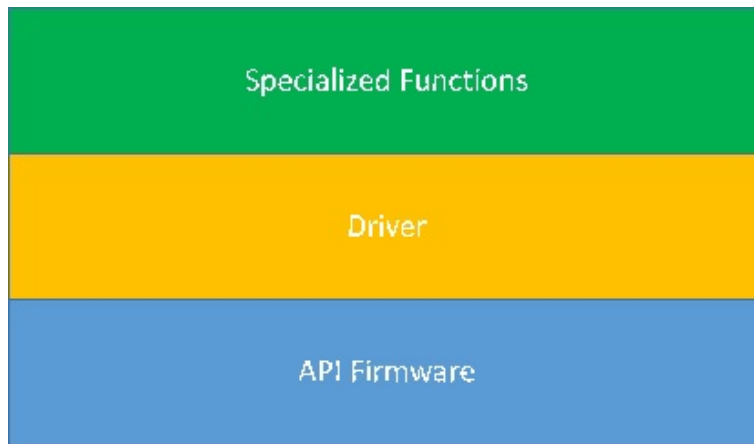


Figure 4.1: *Division of the software in three layers each independent of the other*

The ADCS software is divided into the following layers

1. **API Firmware:** In this layer, all the low-level functionalities are collected. By low-level functionalities, we mean all those functionalities which are not directly relevant to the functioning of the algorithms. This layer doesn't provide any data to the algorithms. Rather, this layer is concerned only with initializing and configuring the various peripherals that we have on the ADCS.

2. **Driver:** This layer acts as a bridge between the lower level API firmware and the higher-level specialized functions that we require to run our algorithms. Ideally, in this layer we find the functions which get the raw data from the sensors, convert or calibrate them as necessary and send them to the algorithms for further use.
3. **Specialized Functions:** The specialized functions are the functions which implement the algorithms for determining and controlling the attitude of the satellite. These functions are high-level functions and require the data provided to them by the *Driver* layer above to function.

4.2 Structure of classes

Once we have abstracted the software as being divided into layers depending on the kind of functions we will write for the software, we consider next the division of the software into dedicated classes. The class structure as envisaged for the ADCS is summarized in the graph below. The arrow head signifies that a particular class gets a particular piece of data from the class where the arrow originates.

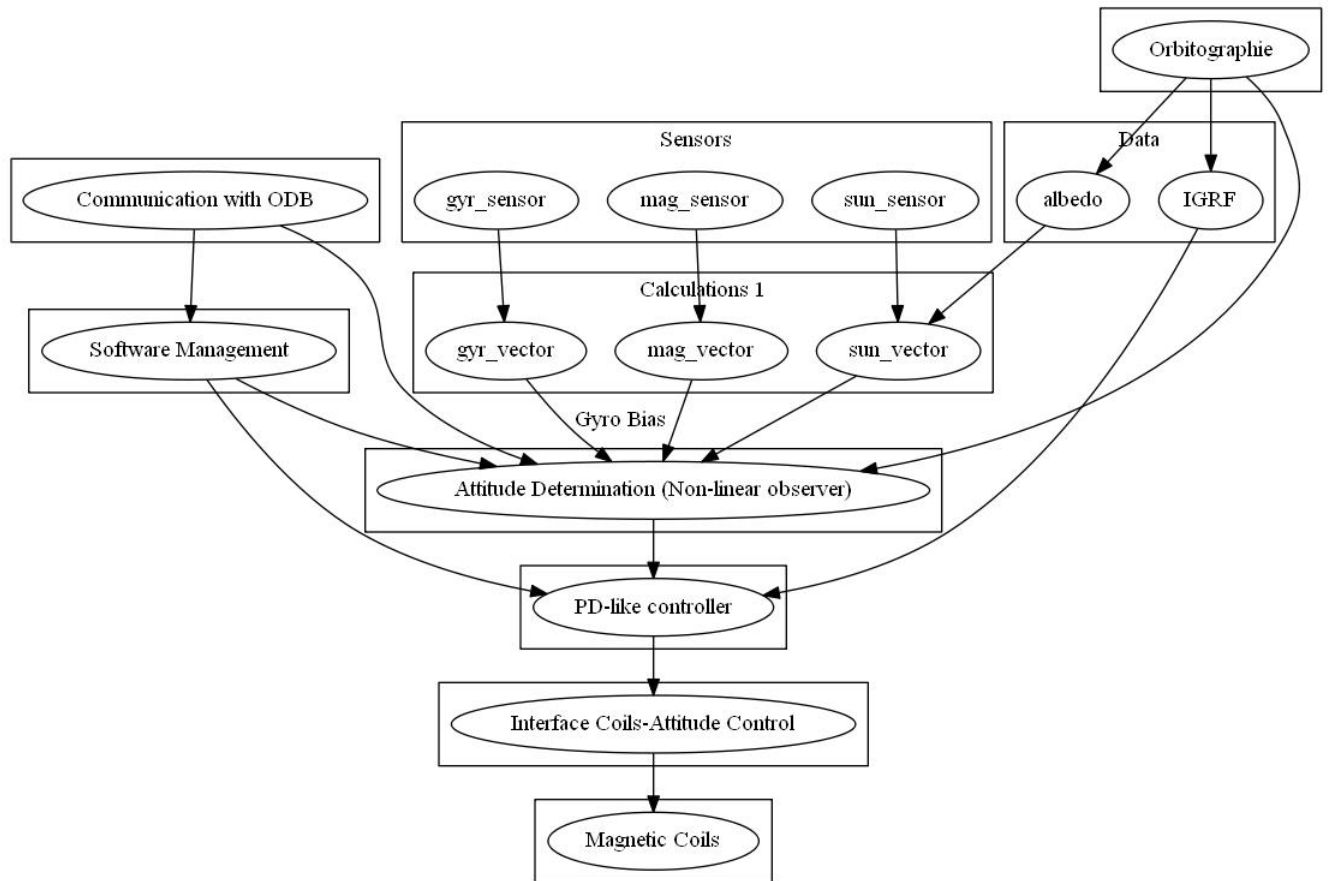


Figure 4.2: Class structure for ADCS Software

The various layers as shown in the class structure above are detailed below

1. **Sensors:** This layer corresponds to the API Firmware layer that we presented earlier. We have three types of sensors: Gyrometer, a magnetometer and 9 sun-sensors. This part of the software corresponds to the initialization and configuring the sensors (and the ports

to which they are associated on the microprocessor) so that they can start generating the data.

2. **Calculations 1:** This layer consists of a first layer of calculations effectuated once raw data is received from the sensors. In this layer, we receive the raw data from the sun-sensors, magnetometer and the gyrometer.
3. **Calculations 2:** This corresponds to a supplementary layer of calculations which is effectuated once the raw data has been received (not shown in the class structure). This step might take different forms for different sensors. For instance, once we have received the sun sensor raw data, the supplementary steps include converting the raw data into a voltage and further combining the data from the 9 sun-sensors into one sun-vector with three components. For the magnetometer however, we can eventually consider calibrating the data with the temperature. Identically for the gyrometer as well.

Finally, this step also relates to reading the terrestrial magnetic field map and the albedo map from the external memory. All this data once collected, calibrated and treated will be given to the algorithms, which we explain below.
4. **Attitude Determination and Control Phase:** Once the data has been generated (after eventual calibrations), we have the layer where the actual control of the satellite takes place. The details of the algorithms will be presented in an ulterior section.
5. **PD-like controller:** This is what controls the satellite by providing us as output the magnetic moment to be generated in the magnetic coils. The controller then gives its output to the Coils-Attitude Control Interface.
6. **Interface Coils-Attitude Control:** This layer corresponds to the calculations to be done to configure the magneto-torquers. The magneto-torquers are controlled by Pulse-Width Modulation(PWM). For PWM, we require the frequency of modulation (in KHz) and the duty cycle (in percent). The result of the algorithms, however, is the magnetic moment to be generated by the magnetic torquers. Thus we require precise data on the behaviour of the magnetic coils used to convert this magnetic moment into a duty-cycle. (The frequency of modulation shall be fixed for all the coils)
7. **Magnetic Coils:** This is the layer which corresponds to the actual control that will be applied via the magnetic coils. The coils are powered via PWM for a fixed frequency usually around 15-20KHz.

4.3 Definition of ports

In this section we define the various ports, what their functionalities are and to which of the peripherals they are connected.

4.4 List of Peripherals

As mentioned earlier, the ADCS disposes the following list of peripherals:

- 32 MB External Memory
- 7(or 9) sun sensors
- Magnetometer
- Gyrometer

SPECIFICATIONS ADCS

Peripheral	Name of the port	No. Port on MPU	Logical State	Alternate Function (AF)	AF Details 1	AF Details 2
Sun Sensor	GS5	PC0	N/A	ADC	ADC.Resolution.8b	Channel10
	GS4	PC1	N/A	ADC	ADC.Resolution.8b	Channel 11
	GS3	PC5	N/A	ADC	ADC.Resolution.8b	Channel 15
	GS2	PB0	N/A	ADC	ADC.Resolution.8b	Channel 8
	GS1	PB1	N/A	ADC	ADC.Resolution.8b	Channel 9
PWM	F1	PB14	N/A	TIM1	Output Compare	
	R1	PB13	0 or 1	N/A		
	PB12	PB12	0 or 1	N/A		
	PB11	PB11	TIM2	TIM2	Output Compare	
	PB10	PB10	TIM2	TIM2	Output Compare	
	R3	PA5	0 or 1	N/A		

Table 4.1: *List of peripherals with their associated ports and alternate functions (AF)*

- H-Bridges to modulate power to the magneto-torquers
- A Serial interface with the On-Board Computer (*Ordinateur de Bord* henceforth, ODB)

Each of these peripherals need to be configured and their outputs need to be specified so that the algorithms used for the determining and controlling the attitude function optimally.

Chapter 5

Data formats

Chapter 6

Testing