REPORT

January 24, 2025

Lab 3

Student: Brice Robert

Promo: 2025

Link

Analysis and design of a space-based embedded system (Eurecom exam of Fall 2021). GRADED: you have a one week delay to send me your report and model by email (ludovic.apvrille A.T. telecom-paris.fr). "until" means until the next Friday, 13:30 - Thales Madrid

0.1 Schedule

Stages	Min	Comments
Reqs	45	6 - 10 reqs
Analysis	45	UCD: 6 - 10 Use Cases, AD, SD
Design	60	4 - 7 blocks
Verification	20	

0.2 Objective

Your objective is to model the **software** of a space-based embedded system.

You have exactly 3 hours to model this system and answer various questions: the time is very short. This means that **you have to make modeling assumptions**. **Keep your diagrams simple and readable**, in particular the analysis diagrams.

Your grade takes into account your report and your models. At the end of the exam, reports (in pdf format) and **models** (in TTool format) **must be sent to me by email**, with Alexia Cepero in cc. The report should contain explanations concerning your models, as well as relevant screen captures of models (e.g., interesting simulation traces, formal verification results).

1 System specification

Again, the system to model is the software running aboard the space-based system described below.

1.1 Description

1.1.1 Overall description

A ground station needs to regularly monitor the safety data of a space-based system: 3D position, temperature, battery level, fuel quantity. For this, a ground station can send, via radio-frequencies, a TC (*TeleCommand*) to the space-based system. Once received by the RF receiver, the software of the space-based system gets the request for information. Data of TCs are ciphered. Once the software has deciphered data, it stores data in an intermediate buffer, and a task to handle this request is triggered. This task builds the answer by reading requested values from sensors. Once the answer packet has been built, it is first enciphered and then sent via a TM (*TeleMetry*) to the ground station, using the RF transmitter.

To ensure that the system does not crash, a microcontroller of this system is dedicated to execute a software task that checks, every 10ms, that all other software tasks of the spacebased embedded system are still responsive. For this, a signal is sent to each task. If some of the tasks have not responded to this signal, then the whole system is restarted, apart from the watchdog: the latter is not expected to crash, apart if the battery is too low to power the microcontroller. Obviously, this watchdog task is of prime importance for this reliability of the system.

Sometimes, while the software system is computing a TM, another TC is received. To avoid redundancy, the TM under construction is canceled: a new TM corresponding to the latest TC is computed and sent.

Last but not least, space-based systems are not well protected against high-energy particles. Such a particle can provoke a bit flip from 0 to 1, or the opposite. The memory is the most sensitive elements of the platform. Therefore, for each block of data the software writes into memory, an error correction code (CRC) of this block has to be computed by the software and stored into memory along with the data block. When this block is read, the corresponding CRC must also be read and checked.

1.2 Assignments

I. Personal work

1. Recopy the following text at the beginning of your report (this is mandatory)

I pledge on my honor that I will not receive any unauthorized help on this exam, that I will not help others in any way on this exam, and that all my answers will be my own personal work.

II. Assumptions

1. Your assumptions should be clear. Do list them in the report: that list might evolve according to the models you make afterwards. Make a clear separation between environment and system assumptions. [2 points]

III. Requirements

1. Create a requirement diagram. [3 points]

IV. Analysis

1. Make a use case diagram. [3 points]
2. Continue the analysis in the form you want: activity diagrams, nominal scenario, error scenarios, . . . : you are free to use the diagrams you want. Of course, the idea here is to show important points of the specification. [3 points]

V. Design and validation

1. Make a block diagram. Put the emphasis on which blocks are used to model the system being designed, and which ones are used either to model the environment, or to prove properties (observers). [2 points]
2. Draw state machines, and provide a nominal simulation trace, as well as an error trace. [3 points]
3. Prove that a TC always result in a TM, apart if another TC is received before sending TM. Last, from requirements, define a property of your choice, and prove whether it is satisfied (or not!). And obviously, explain how you have modeled these two properties [3 points]

2 References

Start	Duration	End	Task
01:30:00	00:45:00	02:15:00	Requirements
02:15:00	00:45:00	03:00:00	Analysis
03:00:00	01:00:00	04:00:00	Design
04:00:00	00:20:00	04:20:00	Verification

Start	Duration	End	Task
09:00:00	00:45:00	09:45:00	Requirements
09:45:00	00:45:00	10:30:00	Analysis
10:30:00	01:00:00	11:30:00	Design
11:30:00	00:20:00	11:50:00	Verification

3 General Comment

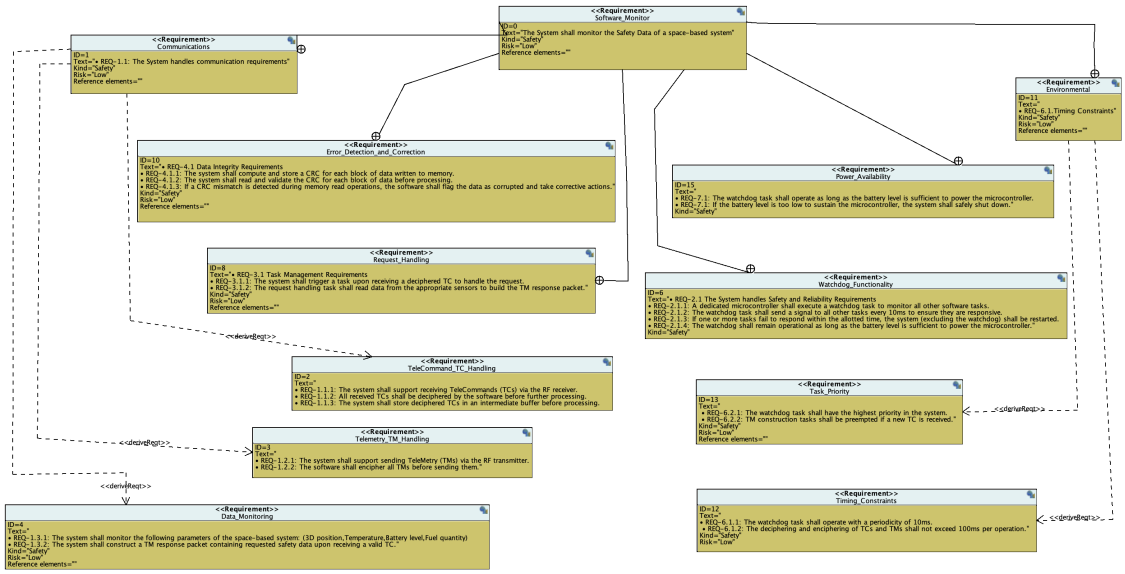
I tried staying in the allocated time to train for the exam. I took to much time in the Requirements and tried speeding up at the end but lacked time to finish.

This is whatever I achieved in the allocated time. The experience helped me in cutting some content for the exam.

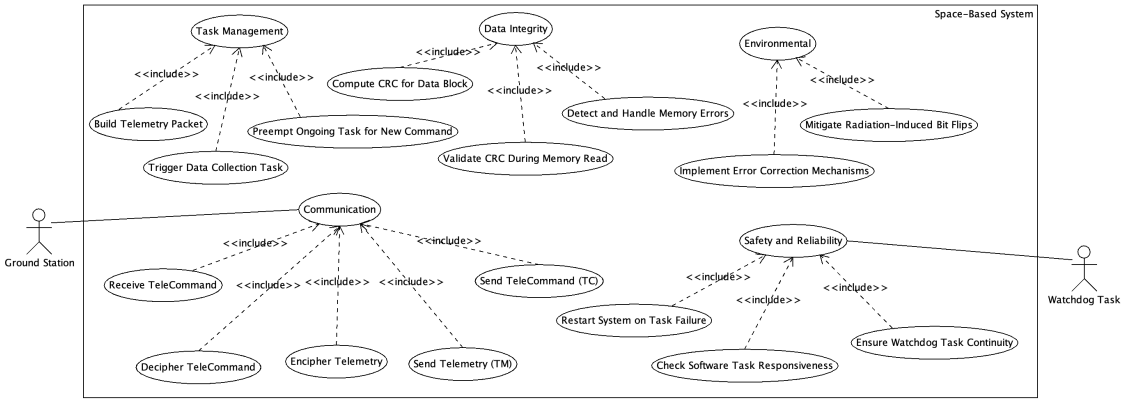
4 Diagrams

The list of the attempted diagrams through the model.

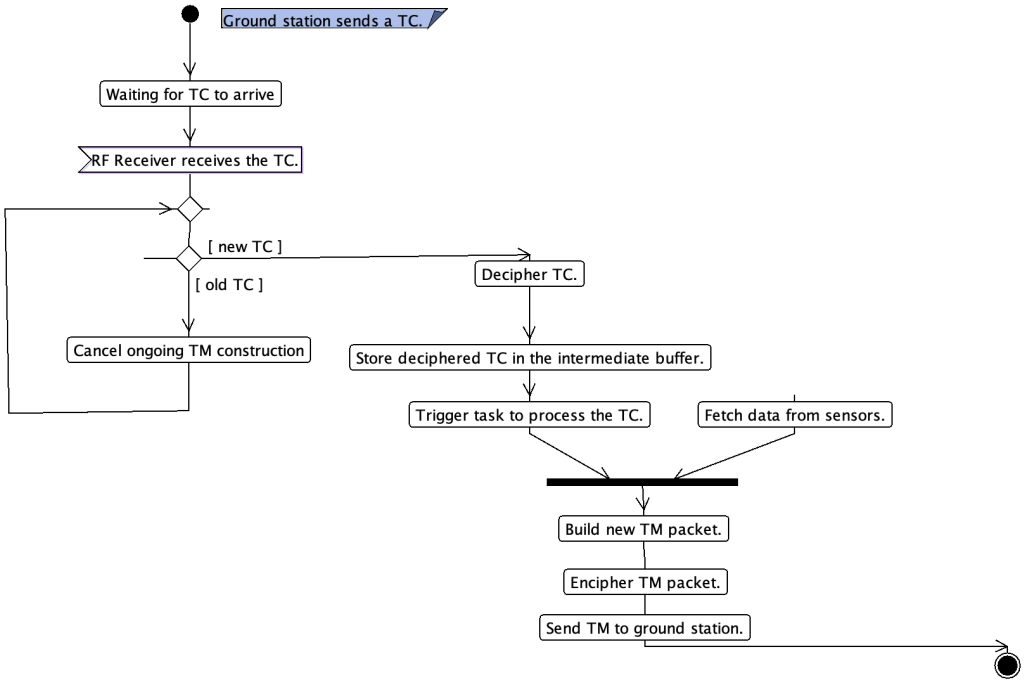
4.1 Requirements



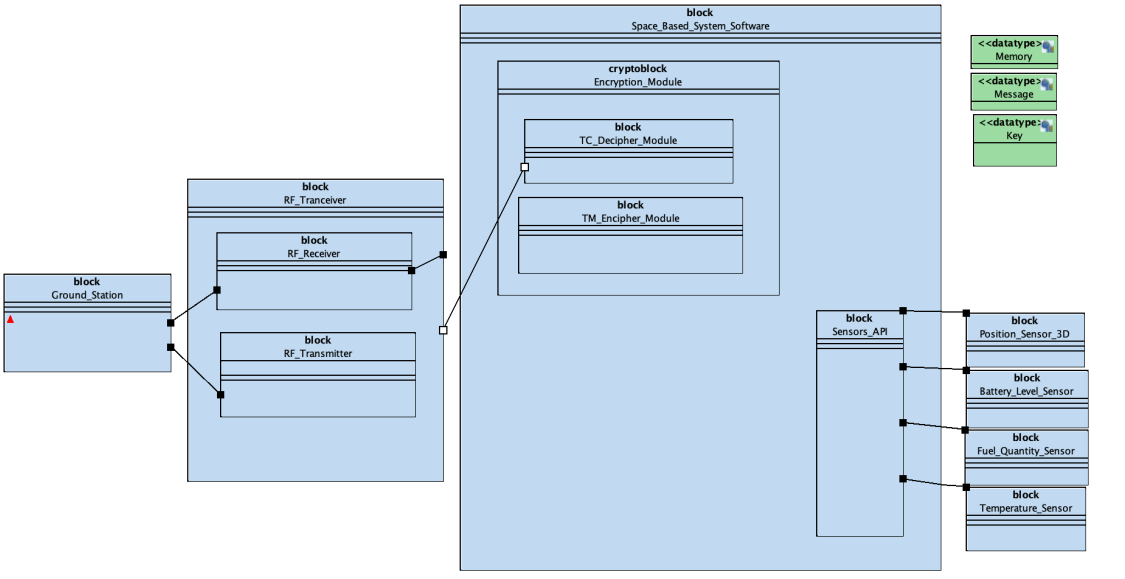
4.2 Use Case Diagram



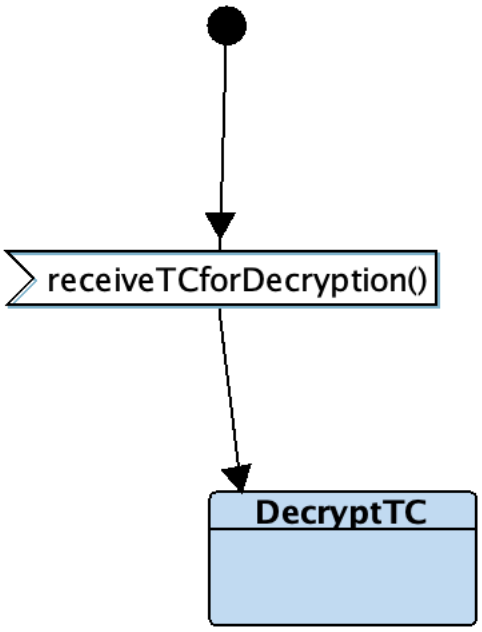
4.3 Activity Diagram

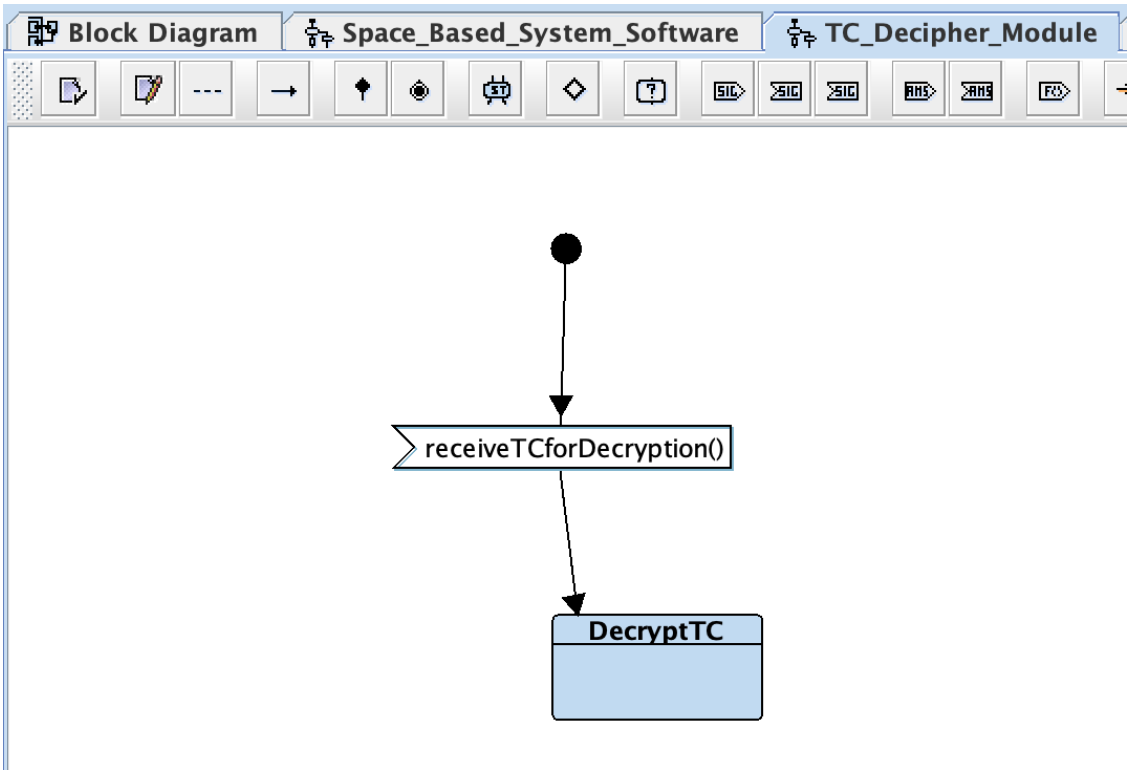


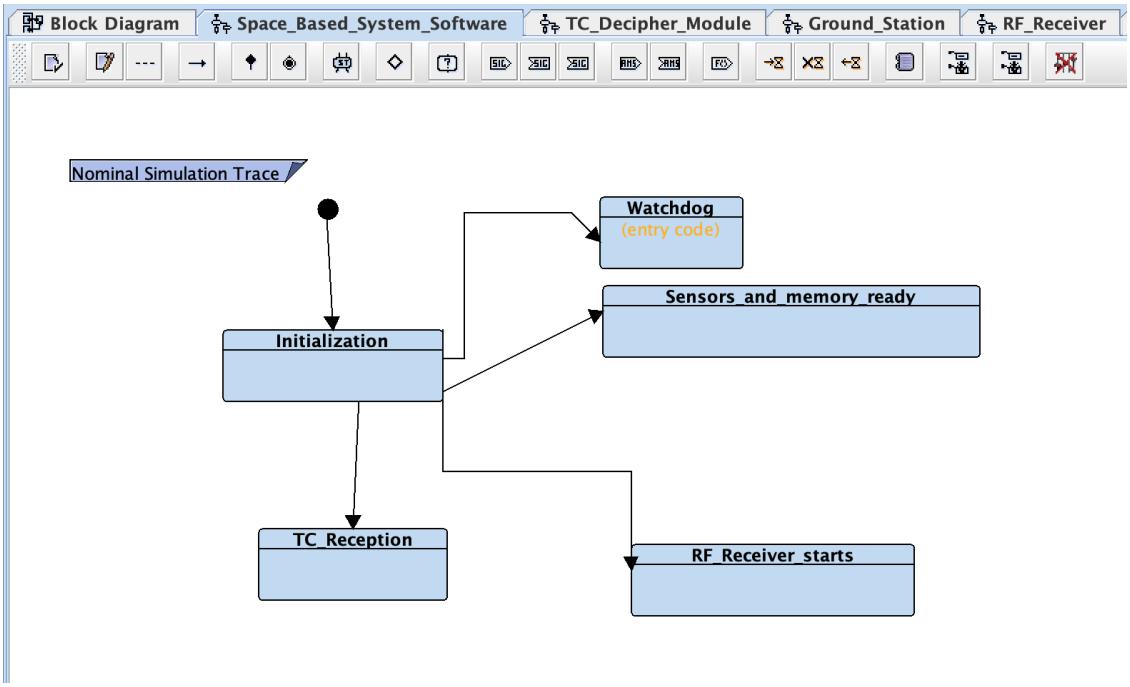
4.4 Block Diagram



4.5 State Machine Diagrams







5 Attempts

> Prove that a TC always result in a TM, apart if another TC is received before sending TM.

Below are the description of the traces I tried to attempt but couldn't due to time.

- Nominal Simulation Trace The simulation of a TeleCommand (TC) lifecycle resulted in:

TC received by RF Receiver → Deciphered → Stored in buffer. Sensor data collected → TM packet built → Enciphered → TM transmitted to the Ground Station. Screenshot: Include a trace showing successful TC-to-TM flow.

- Error Trace A simulation was run to model a memory error:

A high-energy particle flipped a bit in memory. CRC validation failed → Task flagged an error. Watchdog detected the delay → Restarted the system. System reinitialized and resumed normal operations.