Tests :

In order to monitor the evolution of the performance during the different iterations of the project, tests were carried out at the different stages. Bus master scripts are used to test the system. As there are many parameters, many tests are performed. During all the tests, the Wi-Fi connection and the recording of the measurements are activated, as these are the most important parameters and will always be activated when the system is deployed.

The parameters adjusted during the tests are the recording rate and the CAN message rate. As one of the objectives of the project was to add a CAN filter, there are two CAN message rates, one for the messages destined to the telemetry system and another for the other messages on the CAN.

CAN Filter

The zephyr CANBUS API allows a CAN filter to be set. Messages not intended for the telemetry system are also sent over the CANBUS. The system currently used also process these messages, read them and then ignore them. This takes some time and affects the performance of the system. Tests have shown that the system cannot handle more than 200 messages per second to function reliably. By adding a CAN filter, a range can be defined and message ids that are not in the range will be ignored, without affecting the performance of the system.

The CAN filter is set in the initialization of the the CAN device in the *can\_controller.c* file. It takes 2 arguments. An id and a mask. The filter works the following way :  
  
Message is taken if (filter id & mask == message id & mask)

Example :  
id = 0xAA = 0b10101010

Mask = 0xF0 = 0b11110000

Message taken into consideration are 0xA0, 0xA1, … , 0xAF

As the system must be configurable, the id and mask have been added to the configuration file of the system.

"CANFilter":

{

    "id":"0x0",

    "mask":"0x0"

}

Button and Led on the CAN

On the system currently used, the only way to start and stop the recording is to push the physical button. A led on the button indicates if the system is currently recording. This part’s goal is to add a CAN message to start and stop the recording. The telemetry system must also emit a message on the CANBUS to inform if it is currently recording or not. This permits to put the button and the LED on the CANBUS and avoids additional wire for this purpose. With this solution, the Telemetry system will also be able to be controlled by the on-board computer.

The system also needs an option to automatically start at boot. This is in the vision of a new version of the system that automatically records all the time. This is however not possible with only software modifications. The problem is to stop the recording when the system’s power supply is deactivated. Stopping the system when the recording is activated could corrupt the file. To achieve that, the future iteration of the hardware will add a supercapacitor to power up the system during a few seconds after the main power supply is deactivated, to stop the recording. For this part, the hardware must produce a