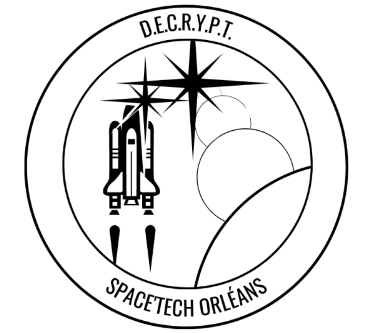


Technical Solutions Overview



Qods ABIDI

Loann KAIKA Rodolphe WEBER

Clément SARAZIN Tuteur

4A GPSE

## Previous project’s solutions

| Solution Technique | Utilisation | Caractéristiques |
| --- | --- | --- |
| LoPy4 (ESP32 Microcontrôleur) | Collecte de données et communication avec le sol via LORA. | Tension : 3.3 à 5.5V, Courant : 133mA, 8 E/S analogiques, Mémoire : 8MB Flash, 4MB RAM, Fréquence : 868MHz, Distance théorique : 46km. |
| Antenne Molex | Amélioration de la portée de communication. | Polarisation : Linéaire, Impédance : 50 Ohms, Gain : 0.3dBi. |
| Pytrack 2.0 (Carte d’extension) | Ajout de fonctionnalités à la LoPy4 (GPS, USB, lecteur SD). | Alimentation : 3.5V – 5.5V, Courant : 14mA, Sensibilité GPS Suivi : -167dBm. |
| MPU9250 (Accéléromètre, Gyroscope, Magnétomètre) | Mesure de l’état de la fusée (accélération, orientation). | Tension : 3 à 5V, Courant : 3.7mA, Gammes : ± 16g (accél.), ± 2000°/s (gyro), ± 4800μT (magnet.). |
| DS18B20 (Capteur de température) | Mesure de la température intérieure et extérieure de la fusée. | Tension : 3 à 5.5V, Courant : 1.5mA, Gamme de mesure : -55 à +125°C ± 0.2°C. |
| MPX4250AP (Capteur de pression) | Mesure de l’altitude basée sur la pression atmosphérique. | Tension : 4.85 à 5.35V, Courant : 7mA, Gamme de mesure : +20 à +250kPa ± 0.35kPa. |
| RP2040 (Microcontrôleur pour OBC) | Contrôle principal de la fusée (décollage, ouverture du parachute). | Alimentation : 1.8 à 5.5V, Processeur : ARM Cortex-M0+ Dual Core, 133MHz. |
| NCP1117 (Régulateur de tension) | Conversion de tension pour alimenter les composants. | Tension d'entrée : 3.5-20V, Tension de sortie : 3.3V, Courant de sortie : 1A. |
| IRF520 (Transistor MOSFET) | Commutation du servo-moteur. | Tension VDS : jusqu’à 100V, Courant IDS : jusqu’à 9.2A. |
| HS-422 (Servo-moteur) | Ouverture de la trappe pour libérer le parachute. | Signal PWM, Courant : 150 mA à 800 mA. |
| W25Q128JVSIM (Mémoire Flash) | Stockage des données du microcontrôleur RP2040. | Alimentation : 2.7 – 3.6V, Taille : 128Mbit, Interface : Quad SPI. |

We decided to do several cards because, according to Planète Science, the sequencer must be separated from any other electronic systems.

## Our solutions

### Seq card

| Solutions techniques | choix | critères de choix | Utilisation | Caractéristiques |
| --- | --- | --- | --- | --- |
| Microcontrôleur | Arduino nano | vitesse 16MHz | assez puissant pour le rôle de l’OBC |  |
| Résistance | 10k |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| Condensateurs |  |  |  |  |

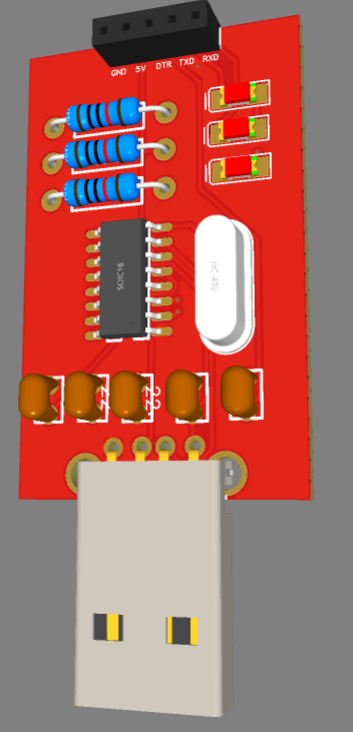
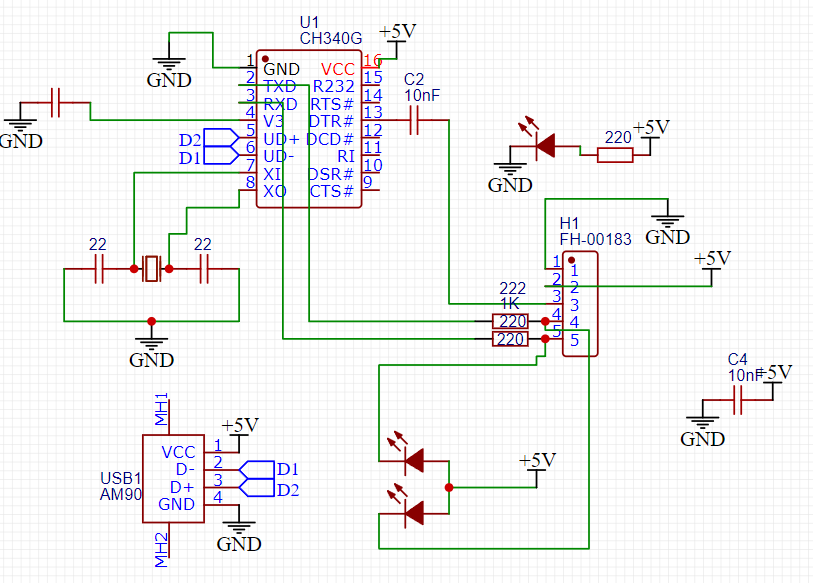
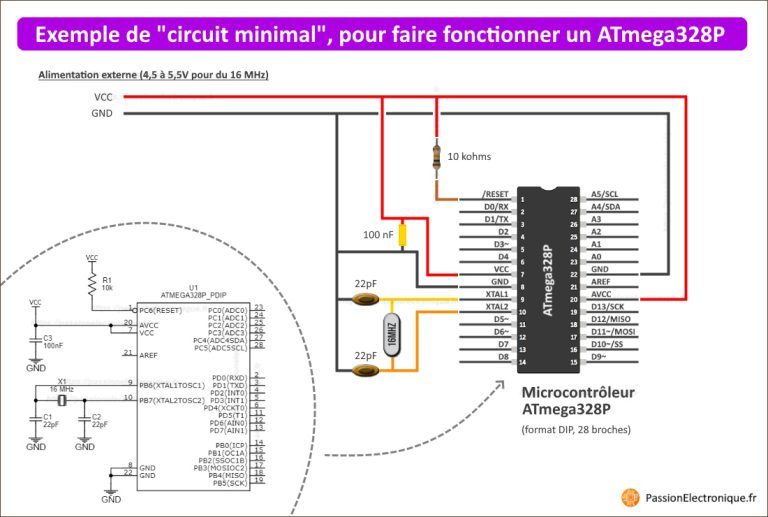
### Power card

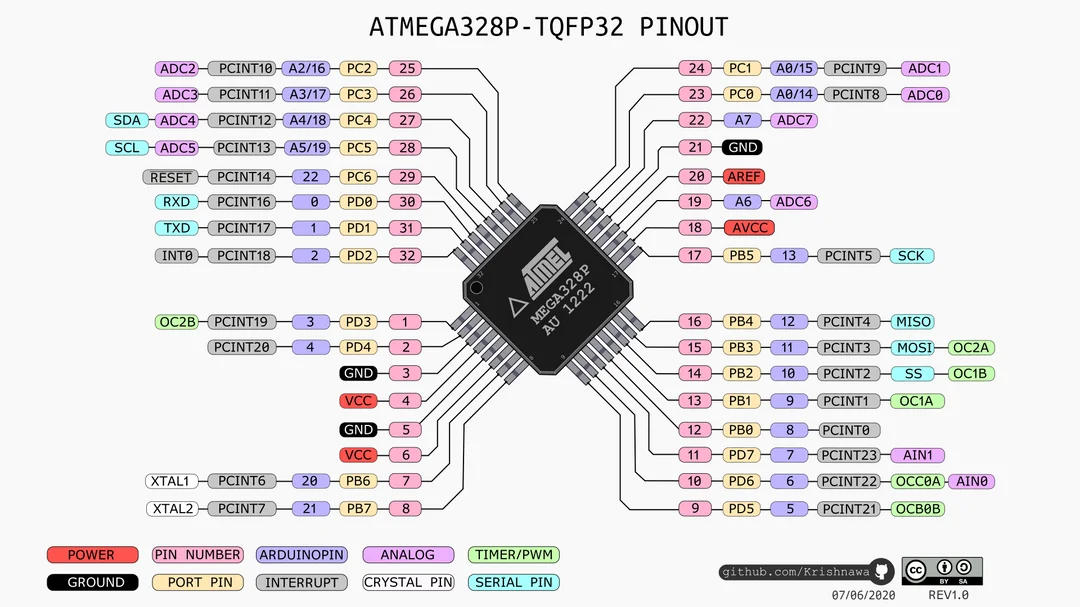
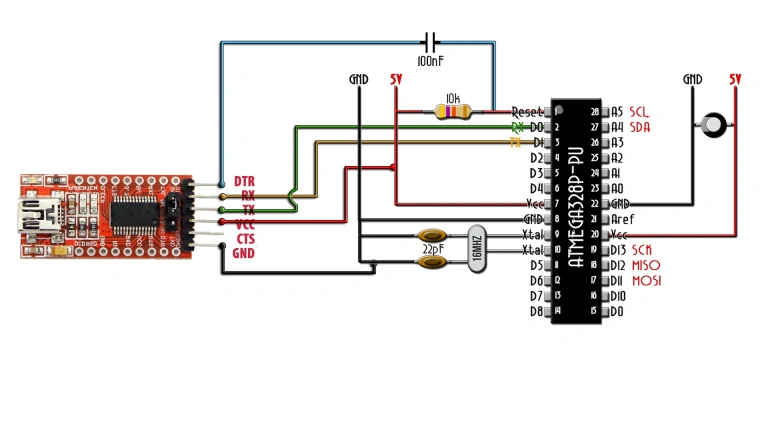
| Solutions techniques | choix | critères de choix | Utilisation | Caractéristiques |
| --- | --- | --- | --- | --- |
| Contrôleur batterie |  |  |  |  |
| Régulateur de tension |  |  |  |  |
|  |  |  |  |  |

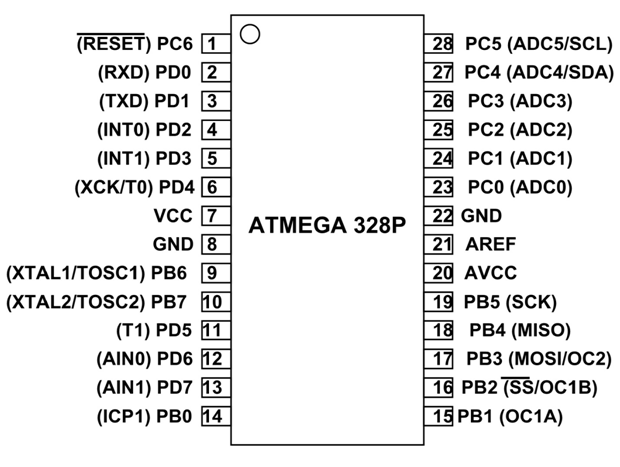
### Telem card

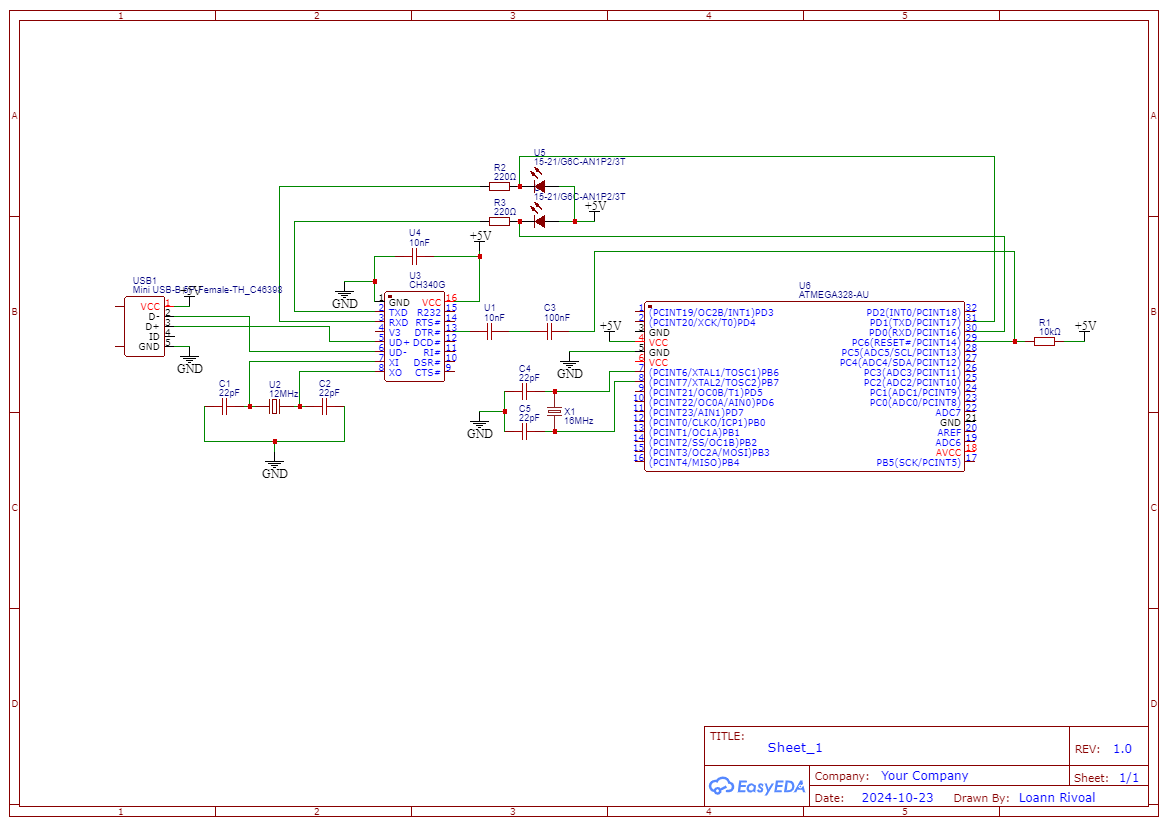
**Raspberry has more GPIO ports than the arduino. And**

| Solutions techniques | choix | critères de choix | Utilisation | Caractéristiques |
| --- | --- | --- | --- | --- |
| micro contrôleur | **Raspberry Pi Pico** | Memory  Speed |  |  |
| gyroscope  magnetometer | MPU9250 | Precision |  |  |
| Module communication | HopeRF  RFM95 |  |  |  |
| pressure sensor | Bosch BMP280 |  |  |  |
| Module gps | Ublox-NEO-7M |  |  |  |

****

****

****

****

| **Feature** | **Specification Sensors** |
| --- | --- |
| **Model** | **GY-91** |
| **Components** | MPU-9250 (9-axis IMU) + BMP280 (barometric pressure sensor) |
| **IMU** | 3-axis gyroscope, 3-axis accelerometer, 3-axis magnetometer |
| **Gyroscope Range** | ±250, ±500, ±1000, ±2000 degrees/second |
| **Accelerometer Range** | ±2g, ±4g, ±8g, ±16g |
| **Magnetometer Range** | ±4800 µT |
| **Barometer Range** | 300 hPa to 1100 hPa |
| **Altitude Resolution** | Approximately 0.1 meter |
| **Communication Protocols** | I²C (default), SPI |
| **I²C Address** | 0x68 (MPU-9250), 0x76 or 0x77 (BMP280) |
| **Power Supply Voltage** | 3.3V to 5V |
| **Current Consumption** | Approx. 5mA |
| **Dimensions** | 13mm × 10mm × 1.5mm (approx.) |

| **Choix**  **Module Stockage** | **Speed** | **Price** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

| **Choix**  **Module GPS** | **Speed** | **Price** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

| **Choix**  **Module communication** | **Speed** | **Price** |
| --- | --- | --- |
| LORA |  |  |
|  |  |  |
|  |  |  |

| **Caractéristique** | **Raspberry Pi Pico** | **Arduino Nano** | **Teensy (4.1)** | **ESP32** |
| --- | --- | --- | --- | --- |
| **Processeur** | RP2040, Dual-core ARM Cortex-M0+ (133 MHz) | ATmega328P (8-bit, 16 MHz) | ARM Cortex-M7 (600 MHz) | Dual-core Xtensa LX6 (240 MHz) |
| **Mémoire Flash** | 2 Mo de flash externe | 32 Ko | 8 Mo | Jusqu'à 16 Mo |
| **RAM** | 264 Ko SRAM | 2 Ko SRAM | 1024 Ko RAM | 520 Ko SRAM |
| **GPIO** | 26 broches GPIO | 14 broches numériques | 55 broches GPIO | Environ 34 broches GPIO |
| **Wi-Fi** | Non | Non | Non | Oui, Wi-Fi 802.11 b/g/n |
| **Bluetooth** | Non | Non | Non | Oui, Bluetooth BLE 4.2/5.0 |
| **ADC** | 3 canaux ADC 12 bits | 8 canaux ADC 10 bits | 2 ADC 12 bits | 18 canaux ADC 12 bits |
| **PWM** | Jusqu'à 16 canaux PWM | 6 canaux PWM | Plusieurs canaux PWM | 16 canaux PWM |
| **Interfaces** | I2C, SPI, UART, USB | I2C, SPI, UART | I2C, SPI, UART, USB, CAN, I2S | I2C, SPI, UART, I2S, CAN, Ethernet |
| **Tension d'alimentation** | 1,8 à 5,5 V | 5 V | 5 V | 3,3 V |
| **Consommation** | Très faible | Faible | Moyenne | Moyenne à faible (selon le mode) |
| **Prix approximatif** | ~5 € | ~15 € | ~25-35 € | ~5-10 € |

### 1. Protocole I2C :

Le GY-91 agit comme un périphérique esclave sur un bus I2C.  
Les trames se composent de plusieurs parties :

| Étape | Contenu de la trame |
| --- | --- |
| Start | Signal de début par le maître (microcontrôleur) |
| Adresse esclave | Adresse du GY-91 (7 bits) + bit de lecture/écriture |
| Commande | Adresse du registre du capteur ciblé |
| Ack | Accusé de réception du GY-91 |
| Données | Données lues ou écrites |
| Stop | Signal de fin de communication |

Adresses I2C courantes :

* MPU9250 : 0x68 (ou 0x69 selon le positionnement du pin AD0)
* BMP280 : 0x76 (ou 0x77 selon la configuration)

Exemple d'échange en I2C :

* Lecture d'une donnée d'accélération sur l'axe X :
  + Envoyer : [0x68, 0x3B] (adresse de l'IMU et registre AX\_XOUT\_H)
  + Réception : Deux octets de données correspondant à l'accélération X.

# 

# **Comparaison des Protocoles de Communication sur Microcontrôleurs**

Cette analyse compare les vitesses de communication I²C, SPI et UART pour quatre microcontrôleurs populaires : Arduino Nano, ESP32, Raspberry Pi Pico, et Teensy 4.1. Les temps de transfert sont également calculés pour des exemples typiques.

| **Carte** | **I²C Max** | **SPI Max** | **UART Max** | **Temps I²C (37 bits)** | **Temps SPI (24 bits)** | **Temps UART (8 bits)** | **Microcontrôleur** | **Fréquence CPU** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Arduino Nano** | 400 kHz | 8 MHz | 1 Mbps | 92.5 µs | 3 µs | 80 µs | ATmega328P | 16 MHz |
| **ESP32** | 3.4 MHz | 80 MHz | 5 Mbps | 10.9 µs | 0.3 µs | 16 µs | ESP32 (Xtensa LX6) | 240 MHz |
| **Raspberry Pi Pico** | 1 MHz | 62.5 MHz | 3 Mbps | 37 µs | 0.38 µs | 26.7 µs | RP2040 (Cortex-M0+) | 133 MHz |
| **Teensy 4.1** | 1 MHz | 120 MHz | 15 Mbps | 37 µs | 0.2 µs | 5.33 µs | i.MX RT1062 (Cortex-M7) | 600 MHz |

## Calculs des Temps de Transfert

Les temps de transfert sont calculés selon les formules suivantes :

· I²C (37 bits) : Temps = 37 bits / Vitesse I²C (en bits/s)

· SPI (24 bits) : Temps = 24 bits / Vitesse SPI (en bits/s)

· UART (8 bits) : Temps = 10 bits / Vitesse UART (en bits/s)

## Exemples de Calculs

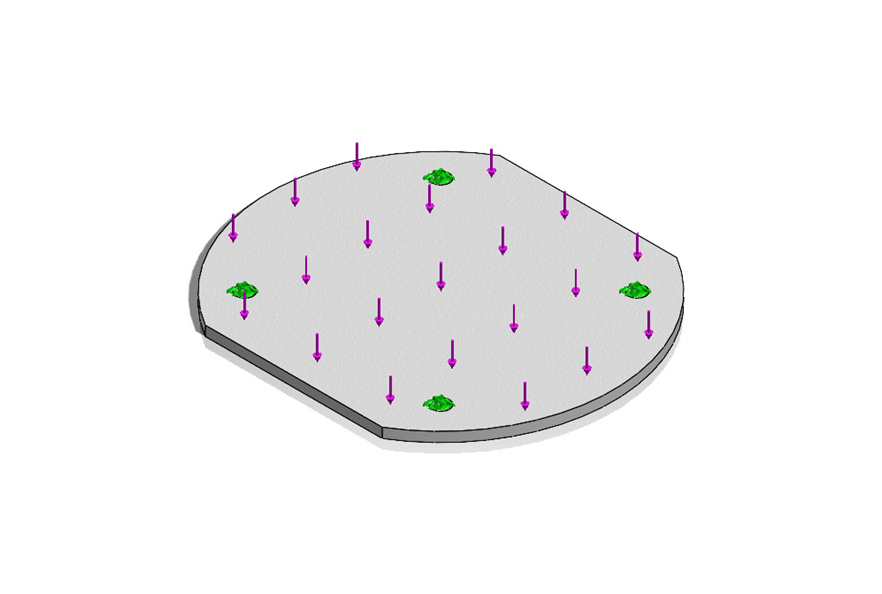
Exemple pour l'Arduino Nano (I²C à 400 kHz) :  
 T = 37 / (400 × 10³) = 92.5 µs

Exemple pour l'ESP32 (SPI à 80 MHz) :  
 T = 24 / (80 × 10⁶) = 0.3 µs

Exemple pour la Teensy 4.1 (UART à 15 Mbps) :  
 T = 10 / (15 × 10⁶) = 0.67 µs

Les cartes PCB sont en général faites à partir de tissu de verre et de résine époxy

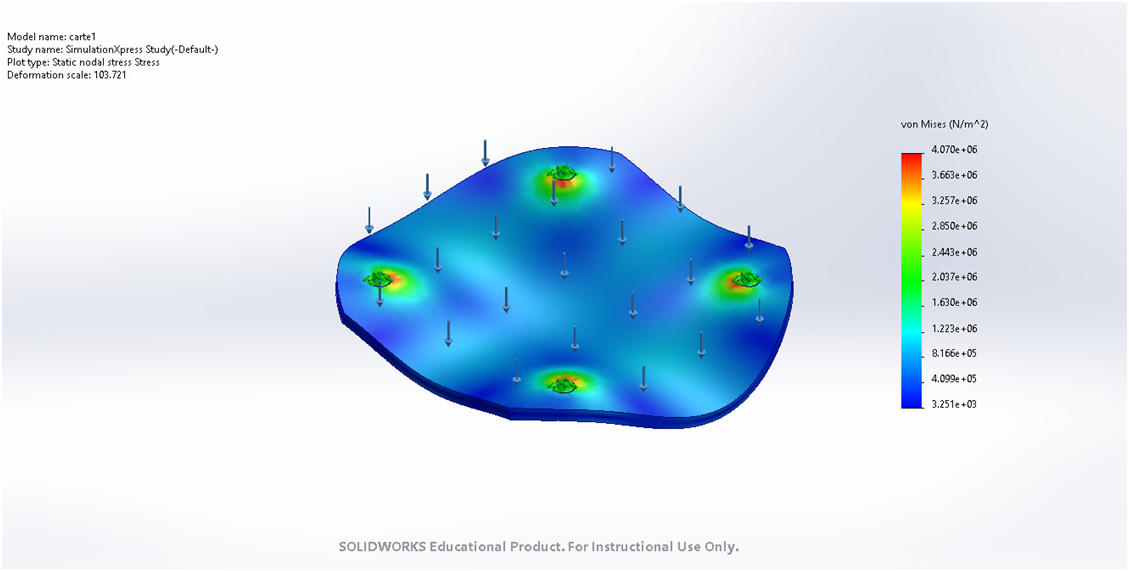
Pour vérifier la résistance de nos cartes, on a effectué des simulations de test HALT (Highly Accelerated Life Test) sur le logiciel Solidworks. On applique une force de 30N équivalent aux vibrations maximum auquel le système fera face en vol.



| Loads and Fixtures  | Fixture name | Fixture Image | Fixture Details | | --- | --- | --- | | Fixed-1 |  | | **Entities:** | **4 face(s)** | | --- | --- | | **Type:** | **Fixed Geometry** | |      | Load name | Load Image | Load Details | | --- | --- | --- | | Force |  | | **Entities:** | **1 face(s)** | | --- | --- | | **Type:** | **Apply normal force** | | **Value:** | **30 N** | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

# **Study Results**

| **Name** | **Type** | **Min** | **Max** |
| --- | --- | --- | --- |
| Stress | VON: von Mises Stress | 3.251e+03N/m^2  Node: 3185 | 4.070e+06N/m^2  Node: 40 |
| carte1-SimulationXpress Study-Stress-Stress | | | |



| **Name** | **Type** | **Min** | **Max** |
| --- | --- | --- | --- |
| Displacement | URES: Resultant Displacement | 0.000e+00mm  Node: 1 | 6.749e-02mm  Node: 3185 |

On voit qu’il y plus de contraintes appliquées sur les fixations donc il faut renforcer cette partie en ajoutant des vias métallisés.