



MINISTÈRE
DE L'ÉDUCATION
NATIONALE

EAE SIN 3

SESSION 2019

AGRÉGATION CONCOURS EXTERNE

Section : SCIENCES INDUSTRIELLES DE L'INGÉNIEUR

Option : SCIENCES INDUSTRIELLES DE L'INGÉNIEUR
ET INGÉNIERIE INFORMATIQUE

CONCEPTION PRÉLIMINAIRE D'UN SYSTÈME,
D'UN PROCÉDÉ OU D'UNE ORGANISATION

Durée : 6 heures

Calculatrice électronique de poche - y compris calculatrice programmable, alphanumérique ou à écran graphique – à fonctionnement autonome, non imprimante, autorisée conformément à la circulaire n° 99-186 du 16 novembre 1999.

L'usage de tout ouvrage de référence, de tout dictionnaire et de tout autre matériel électronique est rigoureusement interdit.

Si vous repérez ce qui vous semble être une erreur d'énoncé, vous devez le signaler très lisiblement sur votre copie, en proposer la correction et poursuivre l'épreuve en conséquence. De même, si cela vous conduit à formuler une ou plusieurs hypothèses, vous devez la (ou les) mentionner explicitement.

NB : Conformément au principe d'anonymat, votre copie ne doit comporter aucun signe distinctif, tel que nom, signature, origine, etc. Si le travail qui vous est demandé consiste notamment en la rédaction d'un projet ou d'une note, vous devrez impérativement vous abstenir de la signer ou de l'identifier.

Tournez la page S.V.P.

A

INFORMATION AUX CANDIDATS

Vous trouverez ci-après les codes nécessaires vous permettant de compléter les rubriques figurant en en-tête de votre copie

Ces codes doivent être reportés sur chacune des copies que vous remettrez.

Concours	Section/option	Epreuve	Matière
EAE	1417A	103	1268

Total 35 pages : 14 pages sujet-questions + 12 pages DT+ 7 pages DR.

Documents techniques :

- DT1 : InSight instruments
- DT2 : Single Event Upset
- DT3 : Différents types d'architectures de traitement
- DT4 : Documentation technique LTC2499
- DT5 : Extrait de commandes utilisées par l'E-box pour communiquer avec le SEIS
- DT6 : Exemple de structure d'un trame CIP échangée
- DT7 : Extrait code socket en C
- DT8 : Code de Hamming
- DT9 : Code C UART - Fonctions C pour piloter L'UART du FPGA
- DR1 : Diagramme de classes à compléter
- DR2 : Spectrogramme à compléter
- DR3 : Code VHDL Mae UART à compléter
- DR4 : Extrait Code C à compléter
- DR5 : Format du type de trame échangée entre C&DH <-> E-BOX à compléter
- DR6 : Chronogramme à compléter

Sources :

- <https://insight.cnes.fr/>
- <https://insight.jpl.nasa.gov>
- https://netlander.cnes.fr/fr/NETLANDER/Fr/GP_mission.htm
- Équipe Philippe Laudet CNES Toulouse

Conseils aux candidats :

Les différentes sous-parties du sujet sont indépendantes.

Une lecture attentive de l'ensemble du document est conseillée avant de composer.

La présentation du code doit respecter les mots clés du langage cible ainsi que l'indentation des structures algorithmiques.

LA MISSION INSIGHT ET LA CONTRIBUTION FRANÇAISE

Le programme spatial InSight¹ de la NASA a abouti au lancement d'un Lander en mai 2018. Sa mission est d'étudier la structure interne de la planète Mars et d'essayer de répondre à la question de la compréhension des processus qui ont façonné les planètes rocheuses du système solaire, y compris la Terre, il y a plus de quatre milliards d'années (Cf. DT1).

A l'aide d'instruments géophysiques, InSight regarde très loin sous la surface de Mars, détecte les empreintes des processus de formation de la planète et mesure les « signes vitaux » de la planète :

- son « pouls » (sismologie),
- sa « température » (sonde de flux de chaleur),
- ses « réflexes ».

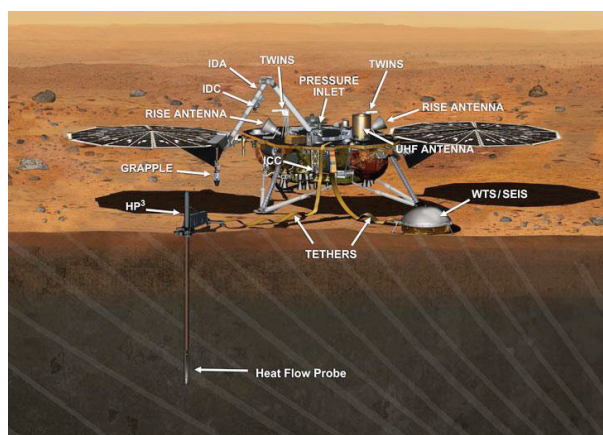


Figure 1 : InSight et ses instruments dans l'environnement Martien

La figure 1 permet d'identifier les différentes expérimentations embarquées. La nomenclature des différents instruments est détaillée dans le DT1.

La charge scientifique est composée de deux instruments :

- l'instrument (SEIS)² de mesure des phénomènes sismiques
- l'instrument (HP³)³ de mesure des phénomènes calorifiques.

Le SEIS a été livré au JPL (Jet Propulsion Laboratory) de la NASA le 18 juillet 2017. Les tests ont montré le bon fonctionnement de l'instrument incluant les 6 sondes sismiques de 2 types (3 VBB⁴ et 3 SP⁵) en interface avec l'atterrisseur et le logiciel de vol. Chaque sonde inclut aussi des capteurs de température.

Dates clés d'InSight :

- Lancement : le 5 mai 2018
- Atterrissage : le 26 novembre 2018
- Opérations de surface : 728 jours / 708 sols (sol = jour martien = 24h39mn35s)
- Déploiement de l'instrument : environ 60 sols (dont une marge de 20 sols)
- Volume de données sur 1 année martienne : plus de 29 Gb (données sismiques traitées publiées sur le Web en 2 semaines, données scientifiques restantes moins de 3 mois, pas de période de propriété).

¹ InSight : Interior exploration using Seismic investigations, geodesy and heat transport

² SEIS : Seismic Experiment for Interior Structure. Cet instrument est fourni par le CNES de Toulouse, l'IPGP de Paris, l'ETH à Zurich, le MPS, l'ICL à Oxford et le JPL à Pasadena.

³ HP³ : Heat Physical Properties Package fourni par l'Agence spatiale allemande (DLR)

⁴ VBB : Very Broad Band – Très Large Bande performances : $9 \text{ m}\cdot\text{s}^{-2}\cdot\text{Hz}^{-1/2}$ from 1 mHz up to 10 Hz

⁵ SP : Short Period – Période Courte performances : $< 5\cdot 10^{-8} \text{ m}\cdot\text{s}^{-2}\cdot\text{Hz}^{-1/2}$ from 10 mHz up to 100 Hz

Sources : <https://insight.jpl.nasa.gov>
<https://insight.cnes.fr/>

Une fois sur Mars, le Lander sera piloté depuis le JPL. Il recevra des ordres consistant en des séries d'actions à réaliser tous les jours martiens qui lui permettront d'exécuter des expériences dont les résultats seront télétransmis sur Terre puis analysés par les équipes de chercheurs internationaux participant à la mission. Plusieurs caméras équipent Insight et rendent compte de son fonctionnement en continu.

OBJET DE CETTE ÉPREUVE DE CONCEPTION

C'est dans ce contexte scientifique que le CNES a réalisé et réalise des tests de fonctionnement sur plusieurs modèles de l'instrument SEIS (figure 2).

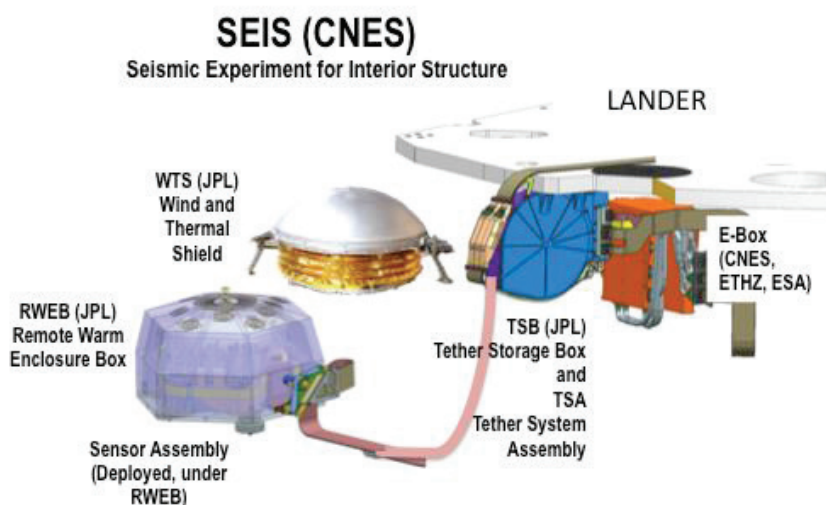


Figure 2 : Ensemble des modules composant le SEIS (source CNES)

Un banc de tests EGSE⁶ (Electronic Ground Support Equipment) a été conçu pour éprouver l'instrument SEIS et la communication de ses mesures au Lander. Ce banc est représenté sur la figure 3 avec l'ensemble des modules impliqués.

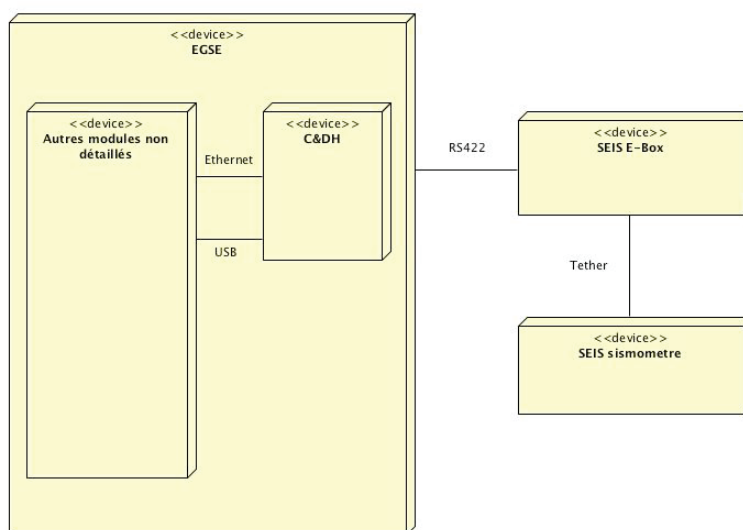


Figure 3 : Architecture des modules du banc de tests

⁶ EGSE : Electronic Ground Support Equipment

Les différents modules de cette partie de l'ESGE sont :

- Le module C&DH : interface de contrôle et de commandes avec le Lander
- Le module E-Box : boîte électronique de contrôle de l'instrument
- Le module Tether : câble de raccordement entre l'E-Box et le sismomètre
- Le module sismomètre : instrument de mesure.

Pour la suite, seul le module C&DH contenu dans le EGSE et les modules E-Box et sismomètre seront retenus pour le questionnement.

Nous nous intéresserons ici à différentes parties de la réalisation du banc de tests ESGE et du sismomètre :

- première partie, choix d'architecture matérielle de l'E-Box,
- deuxième partie, conception de l'architecture logicielle du banc de tests,
- troisième partie, communication entre l'E-Box et le sismomètre,
- quatrième partie, communication entre l'E-Box et le banc de tests,
- cinquième partie, gestion de la mémoire flash,
- sixième partie, cohérence des données acquises.

Première partie : Choix d'architecture matérielle de l'E-Box

L'objectif est de choisir une architecture matérielle pour l'E-Box en précisant, en particulier, les composants principaux à utiliser.

Le système de contrôle, appelé E-Box s'interface avec le banc de tests. Il gère l'ensemble des interfaces de communication tant du côté C&DH que du côté sismomètre et réalise notamment le filtrage des signaux acquis à l'aide de filtres à réponses impulsionnelles finies (ou FIR).

Les figures 4 et 5 fournissent des résultats indiquant les temps de traitement et la consommation énergétique de 5 différentes architectures pour des algorithmes de traitement du signal. Ces algorithmes sont :

- un algorithme de mesure de similarité entre images, appelé **SAD**, pour « sum of absolute differences »
- un algorithme réalisant des convolutions, appelé **Convolution**
- un algorithme de mesure des similarités entre deux variables aléatoires, appelé **Correntropy**.

Les résultats proviennent de l'article « A performance and energy comparison of FPGAs GPUs and multicores for slidingwindow » de Jeremy Fowers and al.

Les architectures considérées sont :

- un CPU simple cœur.
- Un CPU 8 cœurs.
- Un FPGA.
- Un GPU utilisant une librairie d'optimisation pour la FFT (GPU-FFT).
- Un GPU.

QUESTION 1 : En fonction des résultats des figures 4 et 5, classer les architectures qui représentent les meilleurs compromis temps/énergie pour les 3 différents algorithmes. En déduire les deux architectures les plus efficaces pour l'ensemble des algorithmes. Citer au moins 2 critères qui expliquent ces compromis en vous aidant aussi du DT3.

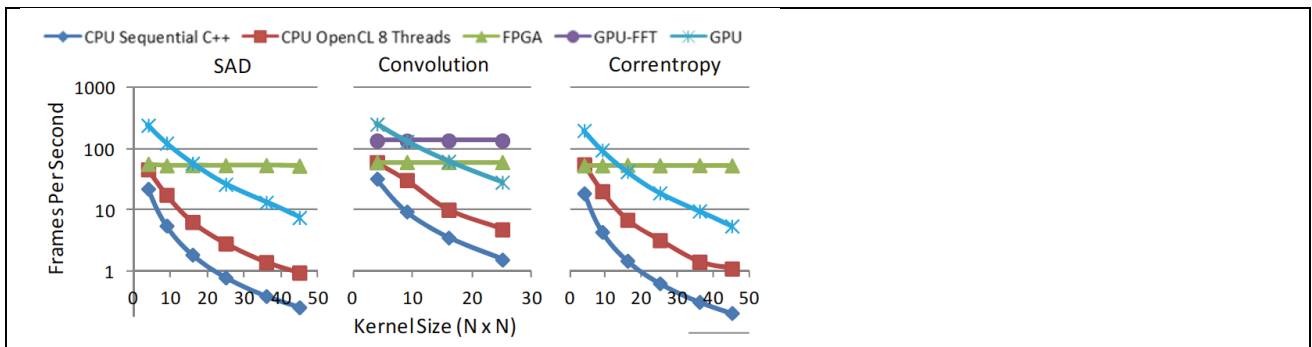


Figure 4 : Temps de traitement de différents algorithmes de traitement du signal sur 5 architectures différentes.

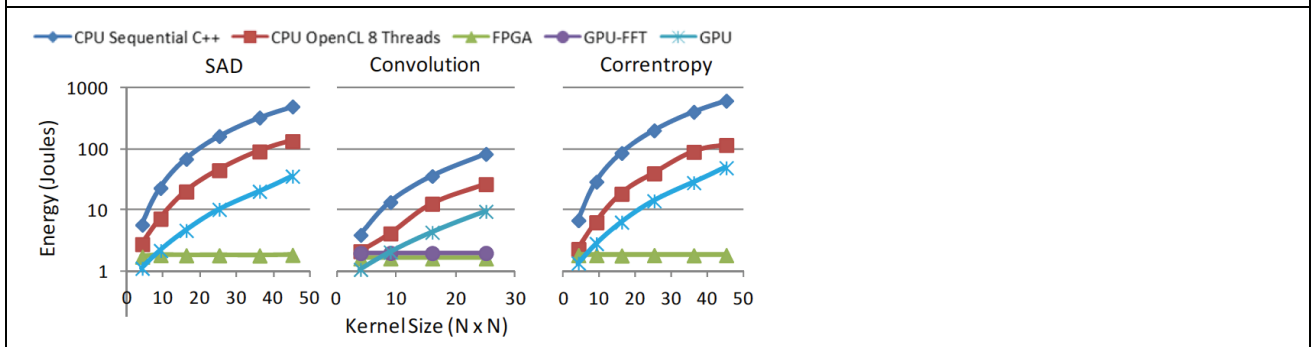


Figure 5 : Énergie dissipée pour différents algorithmes de traitement du signal sur 5 architectures différentes.

QUESTION 2 : En utilisant le DT2, pour les 5 architectures de la question 1, indiquer quels éléments sont les plus sensibles aux rayonnements cosmiques.

En cas de fautes provoquées par des rayonnements cosmiques, il est nécessaire de pouvoir valider les résultats produits par le système embarqué. Il existe deux méthodes pour cela :

- La TMR (Triple Module Redundancy) qui consiste à tripler tout composant et à y associer un voteur majoritaire pour décider de la valeur du résultat
- La reconfiguration dynamique associée à un détecteur de faute. Dans ce cas, si l'apparition d'une faute est constatée, le système est reconfiguré en totalité ou partiellement.

QUESTION 3 : Indiquer pour les 5 architectures de la question 1 quelle méthode est la plus adaptée. Comparer en termes de surface et de temps, ces deux différentes méthodes.

QUESTION 4 : En général, quels sont les avantages et/ou les inconvénients d'un système d'exploitation dans un système embarqué ? En sachant que l'E-Box est sollicitée en continu par le sismomètre qui procède à des mesures régulières à des fréquences d'échantillonnages différentes, mais aussi pour la communication avec le banc de tests, est-il nécessaire d'avoir recours à un système d'exploitation ? Justifier la réponse.

QUESTION 5 : À l'aide du DT3, indiquer comment est réalisée l'interface des entrées-sorties avec un CPU, un GPU et un FPGA.

L'E-Box doit être reconfigurée à distance pour faire évoluer les traitements qu'elle réalise, mais doit aussi satisfaire d'autres contraintes qui sont :

- Une surface utilisée la plus petite possible,
- Une faible consommation,
- Le meilleur coût à performances comparables,
- Un grand degré de liberté sur les entrées-sorties,
- Un temps de traitement respectant une contrainte de latence T_c ,
- Une certaine robustesse aux rayonnements cosmiques.

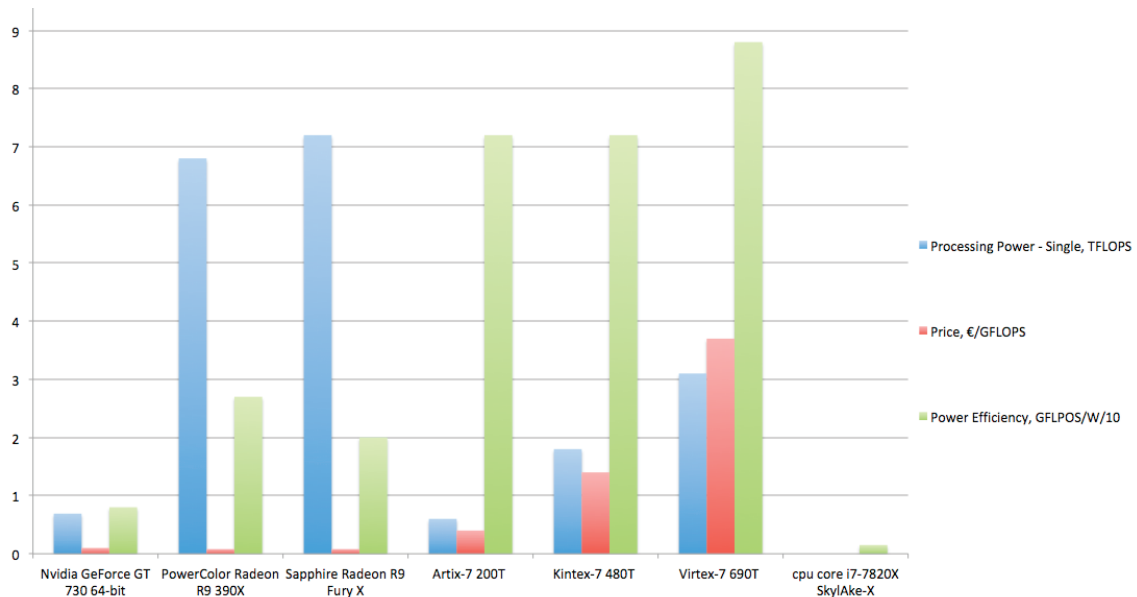


Figure 6 : Caractéristiques techniques de composants de type CPU (Core i7-7820X), GPU (Nvidia GeForce, PowerColor Radeon R9, Saphir Radeon R9) et FPGA (Artix-7, Kintex-7, Virtex-7)

QUESTION 6 : En vous aidant de la figure 6 et des réponses aux questions 1 à 5, quel choix de composant semble le plus adapté pour être intégré dans l'E-Box ? Justifier la réponse.

L'objectif est de compléter l'architecture logicielle du banc des tests. Voici le diagramme de l'architecture logicielle de ce banc de tests.

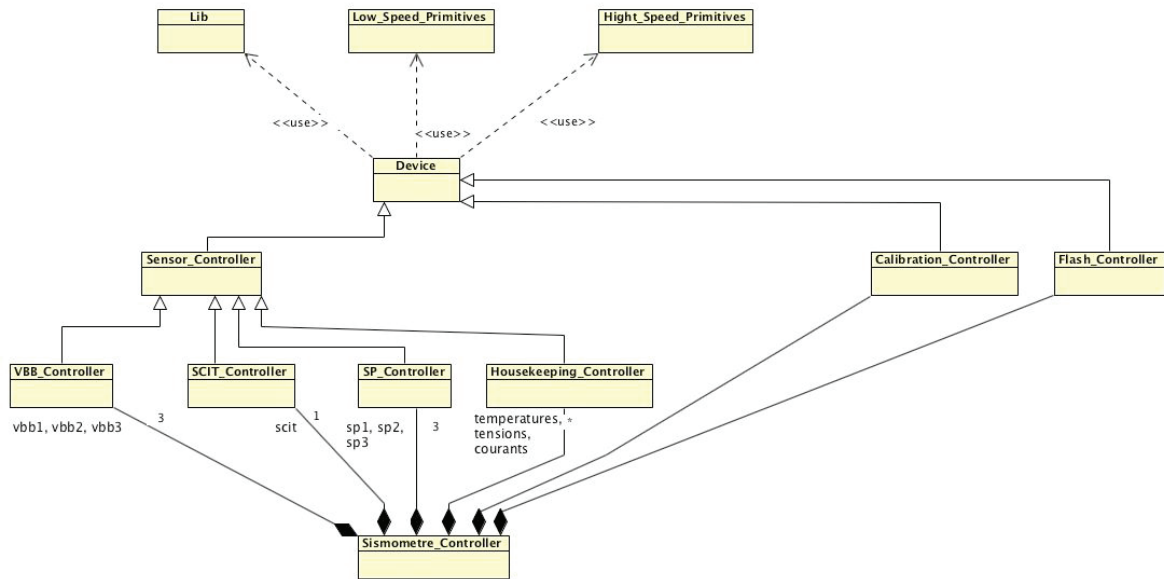


Figure 7 : Diagramme d'architecture logicielle partiel

QUESTION 7: En utilisant la figure 7, indiquer à quoi correspond la classe Device ? Pourquoi mettre un lien de composition entre le SP_Controller et Sismometre_Controller ?

QUESTION 8 : La méthode Seis_Read_Housekeeping() devrait se trouver dans quelle classe ? Justifier. En tenant compte des figures 3 et 7, pourrait-on ajouter une classe Communication_Serie ? Expliquer quel pourrait être son rôle ? Proposer une modification du diagramme de la figure 7.

QUESTION 9 : Expliquer les relations entre les classes Sensor_Controller et SP_Controller ? Que signifie la cardinalité de 3 sur la relation SP_Controller / Sismometre_Controller ?

QUESTION 10 : Indiquer si une instance de VBB_Controller peut accéder à un attribut de la classe Sensor_Controller ?

L'E-Box contient l'essentiel des commandes et contrôles nécessaires au fonctionnement de l'instrument SEIS et l'instrument doit être autonome même quand le reste du lander est au repos. L'E-Box contient plusieurs cartes FPGA, dont certaines contrôlent l'alimentation électrique, l'horloge interne, le mécanisme de stabilisation du sismomètre, la surveillance des capteurs, le système d'acquisition des données provenant des différents capteurs, la mémoire flash pour le stockage des données, les registres des FPGA.

QUESTION 11 : Afin de finaliser la conception de l'architecture logicielle du banc de tests, compléter le diagramme de classes du DR1 en ajoutant les classes de contrôle manquantes.

L'objectif de cette partie est de mettre en œuvre la communication entre l'E-Box et le sismomètre.

L'instrument sismomètre échange des informations avec l'E-Box à travers un câble plat nommé Tether (ce câble permet l'interface avec tous les capteurs et sera posé sur Mars). L'E-Box contient : les cartes de contre-réaction des sismomètres, la carte de pilotage des pieds, la partie convertisseurs (CAN et CNA) et la carte de puissance. Il comporte aussi un certain nombre de capteurs essentiellement analogiques :

- plusieurs capteurs de température type PT1000 (une sonde de température résistive)
- des inclinomètres grossiers (+-0,15° d'étendue de mesure)
- des inclinomètres précis (+-0,1° d'étendue de mesure)
- 3 sismomètres courte période SPx pour mesurer l'accélération selon 3 axes
- 3 sismomètres longue période VBBs pour mesurer l'accélération selon 3 axes inclinés à basse fréquence d'acquisition (<1Hz).

Le banc de test de l'E-Box intégrant le sismomètre, était en fonctionnement sur terre dans sa configuration opérationnelle sur Mars durant la nuit du 11 au 12 février 2018.

Cette même nuit, selon les données du réseau national de surveillance sismique, un séisme de magnitude 4,6 sur l'échelle de Richter est survenu au nord-est de Fontenay-le-Comte, entre la Vendée et les Deux-Sèvres.

QUESTION 12 : Le graphe proposé en DR2 représente les données récupérées par la liaison High speed de l'e-Box durant la nuit du 11 au 12 février. Est-ce que le sismomètre a mesuré ce séisme, si oui, entourer sur le spectrogramme fourni dans le DR2 l'emplacement du séisme et indiquer l'heure de sa survenue (il est demandé ici une précision de 5 minutes). Justifier votre réponse.

Les signaux sont récupérés par l'E-Box et convertis par des CAN pour fournir des valeurs sur 12 bits à 24 bits selon les capteurs. Au sein de l'E-box, les liaisons entre les CAN et les FPGA sont réalisées essentiellement via un bus I2C.

QUESTION 13 : Nous utilisons un CAN pleine échelle dont la documentation est donnée dans le DT4. Indiquer la valeur du bit le moins significatif en sachant que la tension d'alimentation V_{ref} est de 5V. Indiquer la plus petite et la plus grande tension convertible.

QUESTION 14 : À l'aide du DT4, en sachant que nous réglons le CAN avec une vitesse maximale, déterminer si la vitesse de conversion est compatible avec la fréquence d'acquisition des signaux VBBs.

QUESTION 15 : Conclure sur l'adéquation de ce CAN aux besoins de l'E-Box.

L'E-Box utilise des commandes résumées dans le DT5 pour lire les données des mesures : les données Housekeeping (températures, tensions, courants) récupérées par la liaison Low-speed à 1Hz et les données sciences récupérées par la liaison High Speed, celle liée aux capteurs sismiques SP et VBB (Position et Velocity) à 100Hz.

QUESTION 16 : En utilisant le DT5, donner la commande et le code hexadécimal correspondant à envoyer au sismomètre pour lire les différentes données des canaux Housekeeping.

QUESTION 17 : Afin de communiquer avec le FPGA, un UART est réalisé, il est notamment composé d'une machine à états visible sur la figure 8.

Compléter sur le document DR3, le code VHDL de cette machine à états.

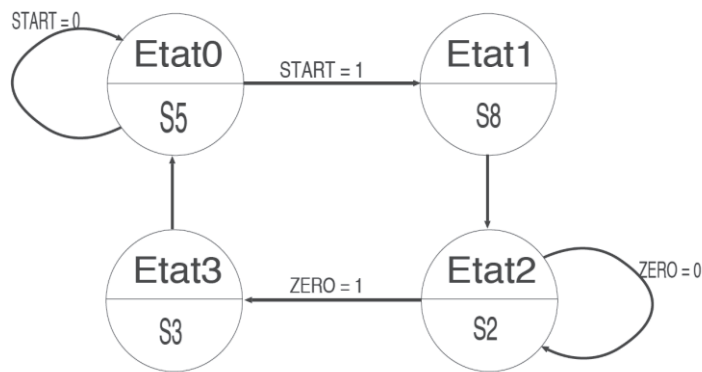


Figure 8 : Machine à états

QUESTION 18 : Pour envoyer une commande entre le FPGA de l'E-Box et les capteurs, un UART est utilisé. Afin de simplifier la gestion de cet UART, un processeur softcore est instancié dans le FPGA. A l'aide du DT9, écrire dans la classe Communication_Serie, une fonction en C qui consiste à initialiser l'UART à 57,6kbaud et à envoyer un mot de 4 caractères sur l'UART du FPGA.

QUESTION 19 : Afin de valider le bon fonctionnement de la communication entre l'E-Box et le sismomètre, indiquer les commandes à transmettre au sismomètre et expliquer pourquoi elles permettent de vérifier le bon fonctionnement de la transmission entre les deux. Utiliser le DT5 pour répondre.

Quatrième partie : Communication entre l'E-Box et le banc de tests

L'objectif de cette partie est de communiquer en utilisant un protocole de haut niveau nommé CIP défini par la NASA.

Les communications sont basées sur le modèle d'architecture réseau illustré par la figure 9.

Les données sont organisées sous forme de trames Ethernet qui encapsule les données de l'instrument dans le datagramme UDP, la couche physique est une liaison RS422.

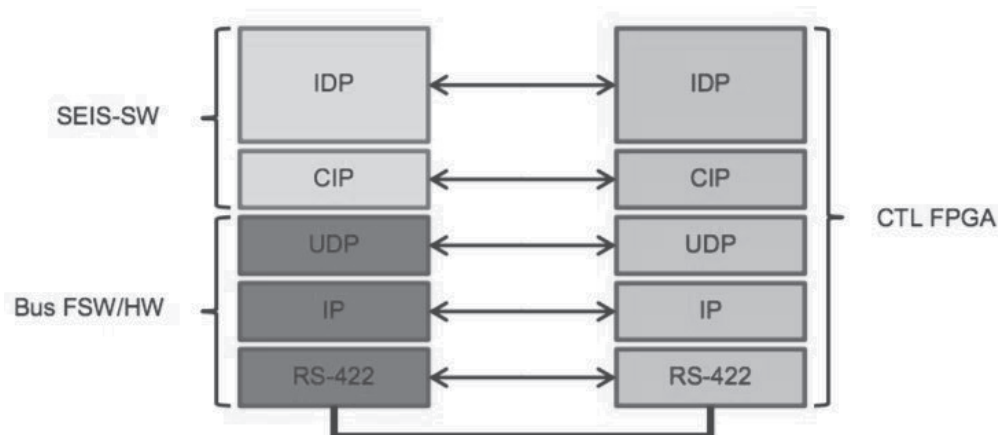


Figure 9 : Modèle d'architecture réseau

Légende :

- RS-422 : Couche physique d'interconnexion électrique standard
- IP (Internet Protocol) : Protocole IPv4 pour les messages à l'intérieur du C&DH
- UDP (User Datagram Protocol) : Pour les messages à l'intérieur du C&DH.
- CIP (Common Instrument Protocol) : Protocole utilisé pour définir les différents types de messages de/vers l'E-Box.
- IDP (Instrument Dependent Protocol) : Ce protocole traite des messages de l'E-Box / C&DH (handshake, ACK/NACK, data protection, etc.).

Le DT6 explique le détail d'une trame échangée sachant que le CIP est un protocole propriétaire du JPL/NASA qui permet d'encapsuler des mesures ou des commandes qui sont échangées avec l'E-Box. Chaque instrument présent sur le Lander communique avec le module C&DH qui lui fera l'interface avec la Terre. Chaque instrument se verra donc attribué une adresse IP.

QUESTION 20 : Expliquer comment est calculé le checksum UDP. Sur quelle partie de trame porte-t-il ?

QUESTION 21 : L'exemple qui est donné dans le DT6 représente un échange entre le module C&DH et l'instrument à travers l'E-Box, déterminer l'adresse IP du C&DH et de l'instrument ? Préciser qui est la source et qui est la destination.

QUESTION 22 : Quelle commande est échangée entre les deux et dans quel sens ?

QUESTION 23 : En utilisant l'extrait de code fourni dans le DR4, ajouter 2 instances de la classe IP_Header_t pour respectivement la trame qui ira du C&DH (contrôleur) vers l'instrument et la trame qui va de l'instrument vers le C&DH (contrôleur). Compléter le DR4.

L'interface avec le Lander se fait avec le module C&DH par des liaisons selon les formats de transmission série suivant la norme RS422.

L'échange Low speed, à vitesse lente, se fait avec le format de transmission de la figure 10.

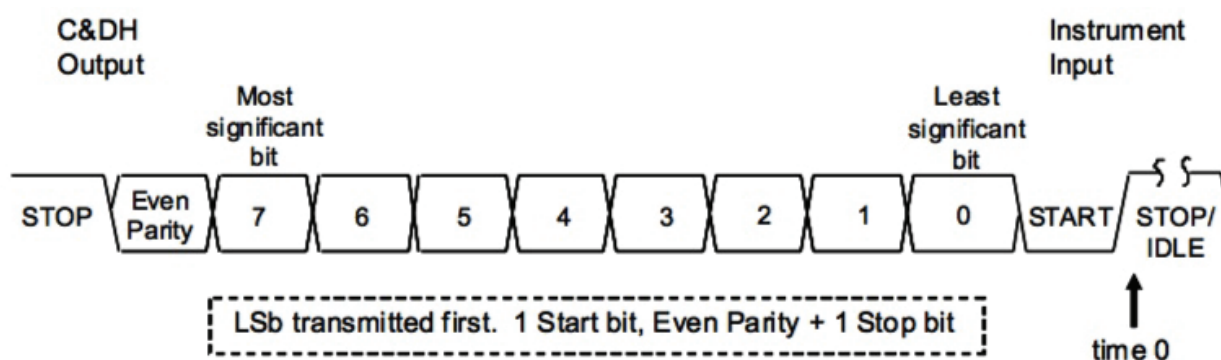


Figure 10 : Format de transmission série des données en Low Speed

Toutes les données sont transmises du bit de poids le plus faible au bit poids le plus fort avec 8 bits de donnée, 1 bit de Start, 1 bit de parité paire et 1 bit de Stop.

La transmission est du même type dans un sens comme dans l'autre.

QUESTION 24 : Quel est l'intérêt principal d'utiliser la liaison RS422 par rapport à une

liaison RS232 dans ce cas de figure ? Le tableau de la figure 11 résume les différences des deux protocoles.

Spécifications	RS-232	RS-422
Fonctionnement	Asymétrique	Différentiel
Nombre d'émetteur/récepteurs sur une ligne	1 émetteur 1 récepteur	1 émetteur 10 récepteurs
Longueur maximale de câble	15,3 mètres	1219,2 mètres
Débit maximum	1 Mbit/s	10 Mbit/s

Figure 11 : Spécifications RS232 et RS422

QUESTION 25 : Quelle est la taille de la donnée transmise pour un octet de charge utile ?

QUESTION 26 : Tout octet qui comprend une erreur de parité est rejeté. Celui-ci n'arrivera donc jamais au JPL aux Etats-Unis. Comment les ingénieurs du JPL peuvent détecter une erreur de parité lors de la transmission sur la liaison RS422 entre l'E-Box et le Lander (C&DH) ?

La figure 12 explique la synchronisation entre les commandes du C&DH et les réponses de l'E-Box sachant qu'aucune donnée n'est renvoyée par l'E-Box sans commande du C&DH.

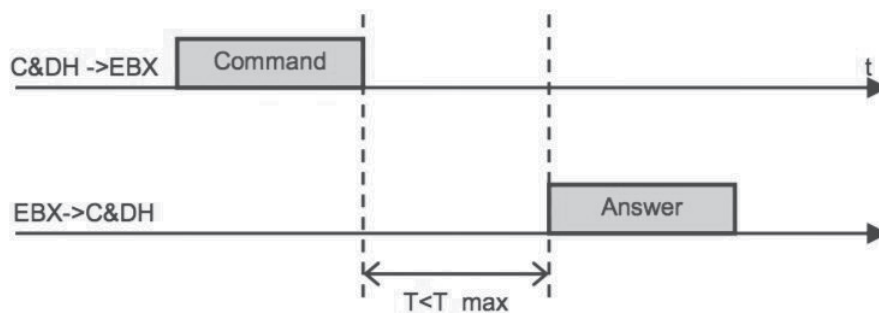


Figure 12 : Protocole de synchronisation des messages entre C&DH et E-Box

(Légende : T = 20ms)

Le débit de la transmission Low-speed est de 57600 bauds. La taille d'un paquet est notée T_p . Sa valeur minimale est de 40 octets et sa valeur maximale est de 1876 octets.

QUESTION 27 : Quelle est le temps de communication théorique en fonction de la taille T_p d'un paquet à partir du début de l'envoi d'une commande par le C&DH jusqu'à la réception complète de la donnée par le C&DH ?

QUESTION 28 : Proposer l'échange de trames qui permettrait d'envoyer une commande « reset » pour initialiser le sismomètre. Combien de trames seront échangées ? Dupliquer si besoin et compléter les formats de trames du DR5.

Au niveau des couches supérieures du réseau, ce sont les sockets à travers des IP de communication intégrées dans le FPGA principal qui sont utilisés. Ces IP peuvent s'interfacer directement avec les horloges FPGA et les broches d'E/S.

QUESTION 29 : Qualifier le socket de l'E-Box ? Dans cette architecture de communication, indiquer qui est client ou serveur. Justifier le mode de fonctionnement.

QUESTION 30 : Utiliser le code fourni en DT7. Préciser et justifier si c'est le code d'un client ou d'un serveur. Que fait ce code ?

QUESTION 31 : Les couches basses du protocole de communication RS-422, IP et UDP sont connues et viennent d'être étudiées. Les couches hautes CIP et IDP sont des couches propriétaires de la NASA. Quels sont les intérêts d'avoir un protocole propriétaires pour les communications de l'E-Box au Lander ?

Cinquième partie : Gestion de la mémoire flash de l'E-Box

L'objectif est de réaliser le stockage des mesures dans la mémoire flash de l'E-Box.

Les mesures provenant du sismomètre et récupérées par le FPGA sont stockées dans une mémoire flash de 4Gbits.

Le FPGA utilise plusieurs mémoires tampons pour transférer les données des paquets en provenance des interfaces séries Low Speed, Housekeeping et High Speed. Un système à base de code Hamming décode et corrige les erreurs. Le code de Hamming permet de détecter et de corriger automatiquement à la réception une erreur qui se serait produite lors de la transmission.

QUESTION 32 : En utilisant le DT8 et sachant que la taille de la donnée en entrée est de 32bits, de combien de bits doit-être le message complet transmis après codage ? Expliquer pourquoi. A quelles places doivent se situer les bits de contrôle de Hamming ?

QUESTION 33 : Pourquoi utilise-t-on le code de Hamming plutôt qu'un autre code dans ce contexte du spatial ? Quel intérêt d'utiliser le bit supplémentaire ?

La mémoire flash est gérée comme un buffer circulaire (FIFO) qui stocke les données en provenance du sismomètre. Elle comporte 4096 pages, chaque page faisant 2112 octets. Elle va contenir les données Sciences (VBB, SP, Temperature(SCIT)). Les échantillonnages se feront pour :

- Les capteurs sismiques SP à 100 Hz sur 24 bits
- Les capteurs sismiques VBB à 20 Hz sur 24 bits
- La température scientifique SCIT à 1 Hz sur 24 bits
- Les données d'Housekeeping (températures, tensions, courants) à 1Hz sur 12 à 16 bits selon les capteurs.

Toutes ces informations sont transmises à la Terre via le Lander sur demande.

Les données SP sont lourdes en raison de la haute fréquence d'échantillonnage, elles ne sont transmises que suite à la détection d'un évènement sismique.

Dans le cas de ces données scientifiques SP, une seconde d'échantillonnage correspond :

- à n échantillons,
- à un paquet,
- à une page.

On considère qu'un bloc mémoire correspond à 64 pages.

QUESTION 34 : Combien d'heures d'acquisition du sismomètre représente les 3711 blocs qui sont alloués pour le stockage des données scientifiques ?

Une action de triplement intervient sur les données critiques qui sont enregistrées sur 3 emplacements en même temps.

QUESTION 35 : En prenant en compte les éléments des questions précédentes, analyser cette solution ? Quelles influences auraient-elles sur les flux de données entre le C&DH et l'E-Box et entre l'E-Box et le C&DH ?

L'objectif est d'explorer des modèles d'acquisition des données pour en assurer la cohérence.

On s'intéresse aux 2 types de données scientifiques VBB et SP reçues par l'E-Box et à l'émission de trames contenant ces données vers le C&HD du point de vue temporel.

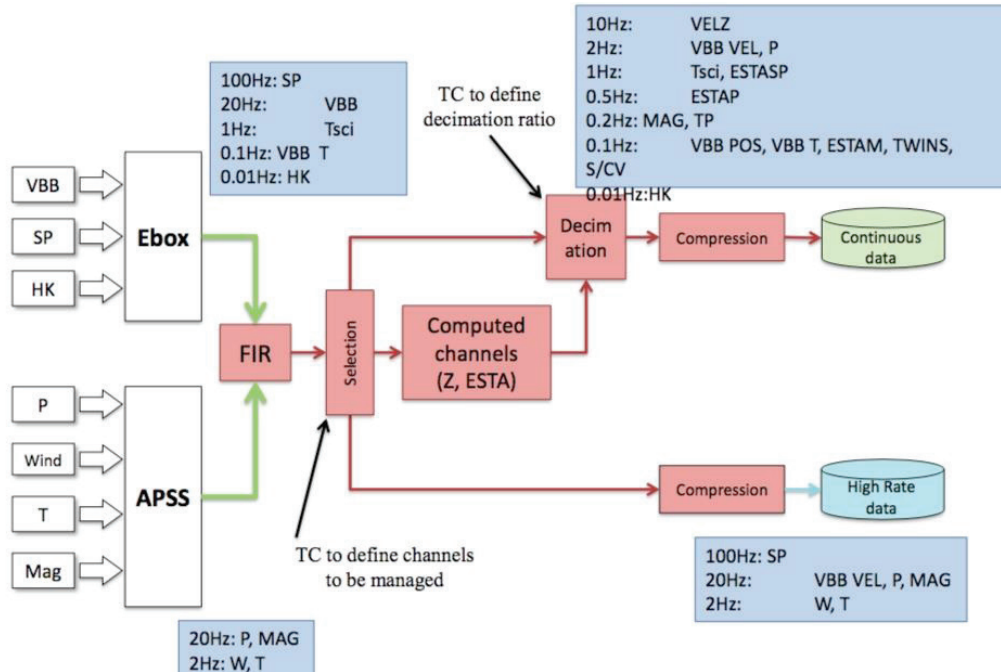


Figure 13 : Présentation P. Lognonné – 6/2014 – Source CNES

La figure 13 précise les fréquences d'acquisition des données. Il est admis ici que le pire-temps d'écriture dans la mémoire « Memory SEIS AC » est inférieur à 1ms pour une donnée de 1024 octets.

QUESTION 36 : Compléter le chronogramme sur le document DR6 qui indique le temps pris pour les écritures des blocs de données d'un VBB et d'un SP sachant que la fréquence d'émission des VBB est de 20Hz, et des SP 100Hz. Nous considérons ici des blocs de données SP et VBB de 2112 octets. Une acquisition SP est lancée à T=0 ms et une acquisition VBB se termine à T=12 ms. Les données sont écrites dans un buffer circulaire de taille fixe dans lequel la donnée la plus récente remplace la donnée la plus ancienne.

L'E-Box comporte une fonction de transfert des données Sciences SP de l'E-Box vers le C&HD. Cette fonction « transfert » consiste à lire les données reçues des capteurs dans la mémoire « Memory SEIS AC », à construire une trame et à envoyer la trame. Les données sont lues dans le buffer circulaire.

La première approche de conception de l'ordonnancement des transmissions par l'E-Box serait de faire la construction de trame de manière répétée sans cadencement. Cela donne alors schématiquement :

```
def transfert_continu() :
    while True :
        transfert()
```

Côté C&HD, il existe une fonction de traitement des données scientifiques qui possède une fréquence de 25Hz. Elle lit les trames reçues et procède à un traitement scientifique.

Soit P1 le processus dans l'E-Box qui exécute la fonction `transfert_continu()`. Soit P2 le processus sur le C&HD qui exécute la fonction de traitement ayant un temps d'exécution pire-cas de 2ms.

QUESTION 37 : Quelles sont les conditions sur le pire-temps d'exécution de la fonction `transfert` pour qu'il y ait eu au moins une nouvelle écriture des données SP d'un bloc de 2112 octets dans la mémoire « Memory SEIS AC » entre deux lectures par la fonction `transfert` ?

QUESTION 38 : Quelles sont les conditions sur le pire-temps d'exécution de la fonction `transfert` pour qu'il y ait eu au moins 3 nouvelles écritures des données SP d'un bloc de 2112 octets?

QUESTION 39 : Quelle est la condition sur le pire-temps d'exécution de la fonction `transfert` pour que la fonction de traitement ait reçue au moins une nouvelle mesure ?

QUESTION 40 : Quelles est la condition sur la taille des données SP, en octet, par rapport au nombre de blocs de 1024 octets du buffer circulaire pour qu'on ne perde pas une mesure entre 2 transferts ?

La seconde approche utilise un trigger qui déclenche la fonction de transfert. On s'assure qu'à toutes les périodes de la fonction de traitement, on démarre la fonction de transfert.

```
def transfert_tt() :  
    rearm_trigger()  
    transfert()
```

QUESTION 41 : A quel moment doit être déclenché le trigger pour assurer que la fonction de traitement récupère ses données ?

Dans la troisième approche, c'est un signal d'interruption qui déclenchera le transfert.

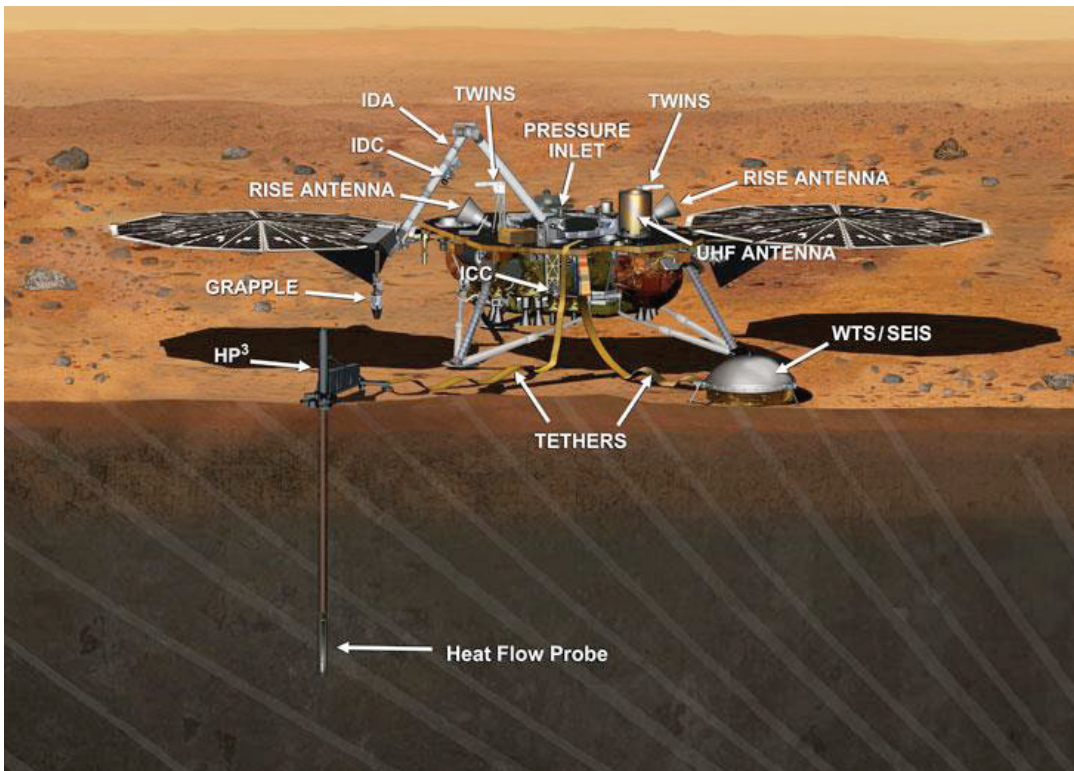
QUESTION 42 : Quel est l'avantage d'un signal d'interruption par rapport au trigger de la seconde approche ? Justifier.

QUESTION 43 : Quelle périphérique d'un système à micro-contrôleur utiliser pour générer ce signal d'interruption ?

En conclusion, analyse :

QUESTION 44 : Comment s'est-on assuré de la cohérence des données ? Expliquer.

Document technique DT1 : InSight instruments



**Artist's
Concept of
InSight
Lander on
Mars
(Annotated)
Date: 22
Dec 2015**

This artist's concept depicts the stationary NASA Mars lander known by the acronym InSight at work studying the interior of Mars. The InSight mission (for Interior Exploration using Seismic Investigations, Geodesy and Heat Transport) is designed to investigate processes that formed and shaped Mars and will help scientists better understand the evolution of our inner solar system's rocky planets, including Earth.

InSight will deploy two instruments to the ground using a robotic arm: a seismometer (contributed by the French space agency Centre National d'Etudes Spatiales, or CNES) to measure the microscopic ground motions from distant marsquakes, providing detailed information about the interior structure of Mars; and a heat-flow probe (contributed by the German Aerospace Center, or DLR) designed to hammer itself 3 to 5 meters (about 16 feet) deep and monitor heat coming from the planet's interior. The mission will also track the lander's radio to measure wobbles in the planet's rotation that relate to the size of its core and will include a camera and a suite of environmental sensors to monitor the weather and variations in the magnetic field. Lockheed Martin Space Systems, Denver, is building the spacecraft.

The following are shown in the annotated image:

- Grapple - Mechanism at the end of the IDA that grips the instruments during deployment
- Heat Flow Probe - Hammering mechanism that pulls the temperature sensors down into the regolith
- HP³ - Heat Flow and Physical Properties Package, the heat flow experiment
- IDC - Instrument Deployment Camera, pointable medium-resolution camera
- IDA - Instrument Deployment Arm
- ICC - Instrument Context Camera, fixed wide-angle camera
- Pressure Inlet - Wind-shielded opening for pressure sensor
- RISE Antenna - X-band radio antenna for the Rotation and Interior Structure Experiment
- SEIS - Seismic Experiment for Interior Structure, the seismometer
- Tethers - Cables carrying electrical power, commands and data between the lander and instruments
- TWINS - Temperature and Winds for InSight, environmental sensors
- UHF Antenna - Antenna used for communication with orbital relay spacecraft
- WTS - Wind and Thermal Shield protecting the seismometer from the environment

InSight is part of NASA's Discovery Program of competitively selected solar system exploration missions with highly focused scientific goals. NASA's Marshall Space Flight Center in Huntsville, Ala., manages the Discovery Program for the agency's Science Mission Directorate in Washington. NASA's Jet Propulsion Laboratory, a division of the California Institute of Technology, Pasadena, manages InSight for the NASA Science Mission Directorate.

Credit: NASA/JPL-Caltech

Source : <http://insight.jpl.nasa.gov>, <http://discovery.nasa.gov>



Document technique DT2 : Single Event Upset

On distingue deux grandes sources de radiations qui influent sur les systèmes électroniques : les radiations qui sont issues directement de notre soleil et les radiations qui proviennent de l'extérieur de notre système solaire. Sur Terre, la ceinture de Van Hallen, voir figure 1, piège une partie de ces rayonnements permettant ainsi à la vie de se développer.

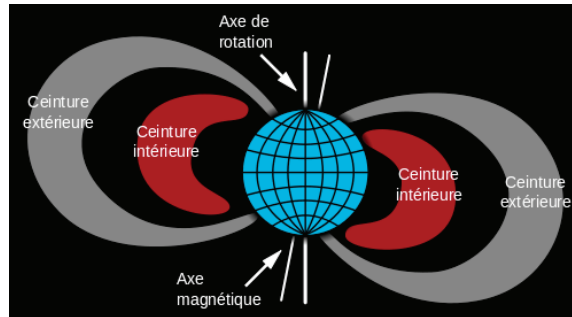


Figure 1 : Ceinture de Van Hallen

Ces sources de radiations génèrent des particules chargées qui peuvent entrer en interaction avec les systèmes électroniques sous forme de SET ou de SEU.

Single-Event Transient (SET) :

Dans un SET, l'interaction produite par la particule se manifeste sous la forme d'une accumulation de charge comme illustré dans la figure 2. La charge électrique ne dure qu'un petit intervalle de temps de quelques picosecondes à quelques nanosecondes.

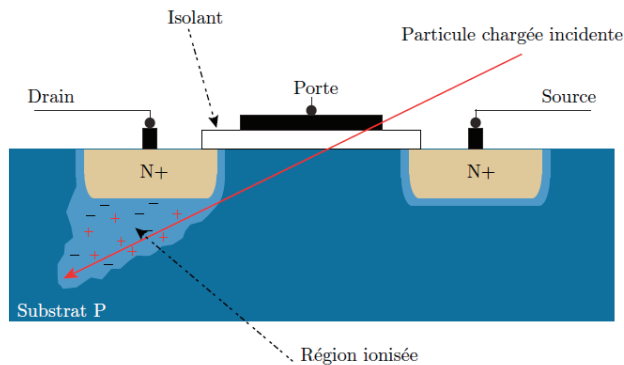


Figure 2 : SET

Single-Event Upset (SEU)

Un SEU peut se produire directement quand une charge déposée est collectée par un élément de mémoire (bascule, verrou, cellule SRAM. . .). Il peut être aussi issu d'un SET qui a été capturé au moment de rafraîchissement de la cellule mémoire. On utilise le terme anglais bit-flip pour désigner la conséquence d'un SEU sur cellule mémoire, et qui signifie l'inversion d'état d'un bit.

Dans la figure 3, une illustration d'un SEU où suite à un SET au niveau du transistor M2, la sortie de l'inverseur, initialement à l'état logique un, est inversée pour représenter après inversion un zéro logique. Cet état perdurera tant qu'il n'y aura pas de réécriture du contenu de la cellule SRAM

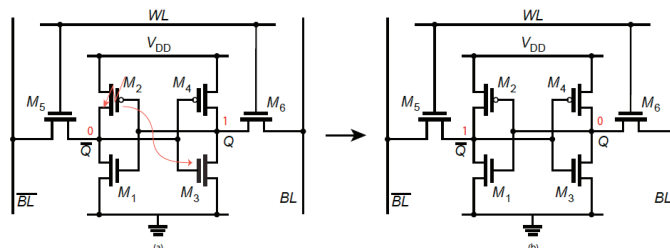
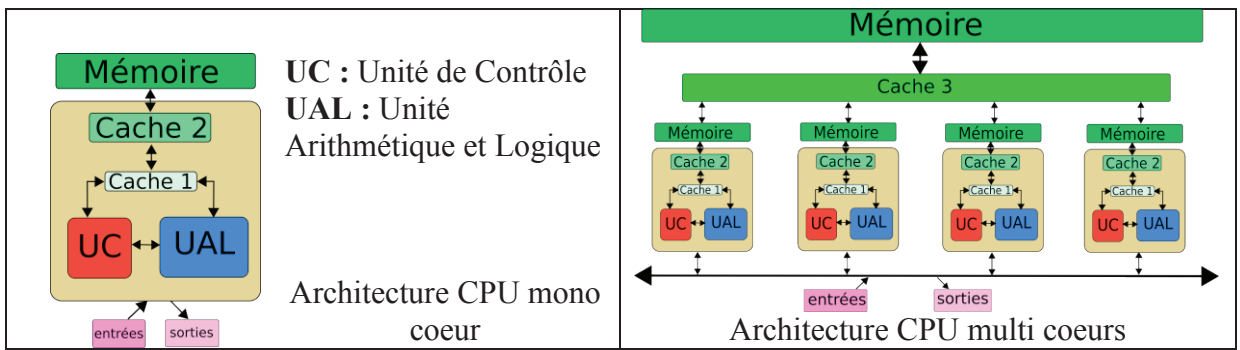
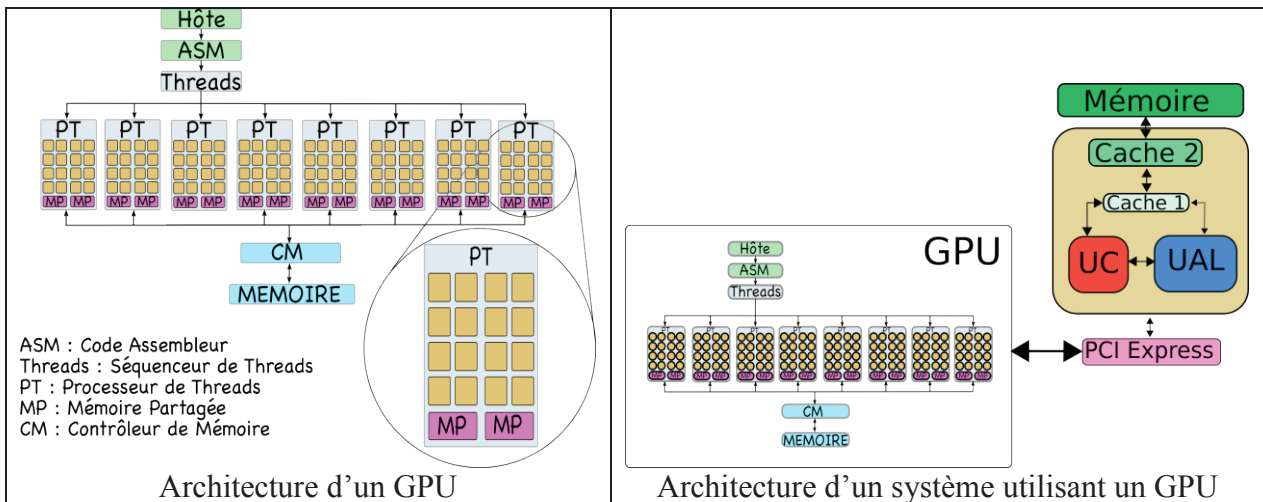


Figure 3 : SEU

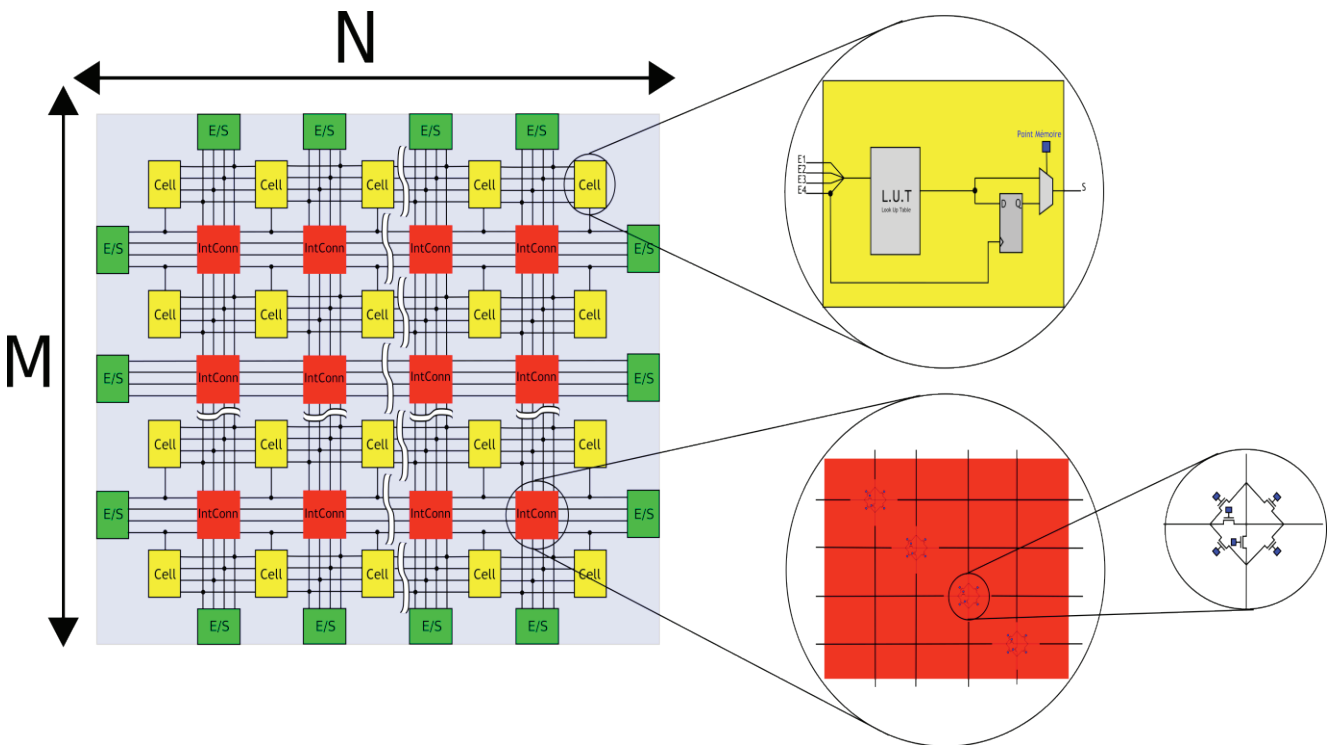
Document technique DT3 : Différents types d'architectures de traitement



1 - Architecture générique d'un CPU



2 - Architecture générique d'un GPU



3 - Architecture générique d'un FPGA



LTC2499

24-Bit 8-/16-Channel
 $\Delta\Sigma$ ADC with Easy Drive Input Current
 Cancellation and I²C Interface

ANALOG INPUT AND REFERENCE The ● denotes the specifications which apply over the full operating temperature range, otherwise specifications are at T_A = 25°C. (Note 3)

SYMBOL	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
IN ⁺	Absolute/Common Mode IN ⁺ Voltage (IN ⁺ Corresponds to the Selected Positive Input Channel)		GND – 0.3V		V _{CC} + 0.3V	V
IN ⁻	Absolute/Common Mode IN ⁻ Voltage (IN ⁻ Corresponds to the Selected Negative Input Channel or COM)		GND – 0.3V		V _{CC} + 0.3V	V
V _{IN}	Input Voltage Range (IN ⁺ – IN ⁻)	Differential/Single-Ended ●	–FS		+FS	V

LTC2499

ANALOG INPUT AND REFERENCE The ● denotes the specifications which apply over the full operating temperature range, otherwise specifications are at T_A = 25°C. (Note 3)

FS	Full Scale of the Input (IN ⁺ – IN ⁻)	Differential/Single-Ended ●	0.5V _{REF}			V
LSB	Least Significant Bit of the Output Code	●	FS/2 ²⁴			
REF ⁺	Absolute/Common Mode REF ⁺ Voltage	●	0.1		V _{CC}	V
REF ⁻	Absolute/Common Mode REF ⁻ Voltage	●	GND		REF ⁺ – 0.1V	V
V _{REF}	Reference Voltage Range (REF ⁺ – REF ⁻)	●	0.1		V _{CC}	V
CS(IN ⁺)	IN ⁺ Sampling Capacitance			11		pF
CS(IN ⁻)	IN ⁻ Sampling Capacitance			11		pF
CS(V _{REF})	V _{REF} Sampling Capacitance			11		pF
I _{DC_LEAK(IN⁺)}	IN ⁺ DC Leakage Current	Sleep Mode, IN ⁺ = GND ●	–10	1	10	nA
I _{DC_LEAK(IN⁻)}	IN ⁻ DC Leakage Current	Sleep Mode, IN ⁻ = GND ●	–10	1	10	nA
I _{DC_LEAK(REF⁺)}	REF ⁺ DC Leakage Current	Sleep Mode, REF ⁺ = V _{CC} ●	–100	1	100	nA
I _{DC_LEAK(REF⁻)}	REF ⁻ DC Leakage Current	Sleep Mode, REF ⁻ = GND ●	–100	1	100	nA
t _{OPEN}	MUX Break-Before-Make			50		ns
QIRR	MUX Off Isolation	V _{IN} = 2V _{p-p} DC to 1.8MHz		120		dB

DIGITAL INPUTS AND DIGITAL OUTPUTS The ● denotes the specifications which apply over the full operating temperature range, otherwise specifications are at T_A = 25°C. (Note 3)

SYMBOL	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
f _{EOSC}	External Oscillator Frequency Range	(Note 16) ●	10		1000	kHz
t _{HEO}	External Oscillator High Period	●	0.125		100	μs
t _{LEO}	External Oscillator Low Period	●	0.125		100	μs
t _{CONV_1}	Conversion Time for 1x Speed Mode	50Hz Mode ● 60Hz Mode ● Simultaneous 50Hz/60Hz Mode ● External Oscillator (Note 10)	157.2 131 144.1	160.3 133.6 146.9 41036/f _{EOSC} (in kHz)	163.5 136.3 149.9	ms ms ms ms
t _{CONV_2}	Conversion Time for 2x Speed Mode	50Hz Mode ● 60Hz Mode ● Simultaneous 50Hz/60Hz Mode ● External Oscillator (Note 10)	78.7 65.6 72.2	80.3 66.9 73.6 20556/f _{EOSC} (in kHz)	81.9 68.2 75.1	ms ms ms ms

Document technique DT5 : Extrait de commandes utilisées par l'E-box pour communiquer avec le SEIS (sources CNES)

GENERAL MESSAGES DEFINITION

Command	Command Identifier	Description	Associated answer on the low speed interface
seis_ping (= NOOP in AD1)	0x01	Sends a NOOP command	ACK_MSG/NACK_MSG
seis_reset	0x02	Acts as the analog reset line on Ebox. Resets all Ebox FPGA registers to the default values	ACK_MSG/NACK_MSG
seis_get_error_flags	0x0C	Asks to retrieve all the Ebox error flags	ACK_seis_get_error_flags / NACK_seis_get_error_flags
seis_reset_error_flags	0x0D	Asks to reset any error flag (by defining it with an argument "mask")	ACK_MSG/ NACK_seis_reset_error_flags
seis_get_acq_status	0x13	Requests the status of the acquisition controller flags	ACK_seis_get_acq_status / NACK_seis_get_acq_status
seis_read_hk	0x3D	Asks to retrieve HK values on the low speed interface (last scan)	ACK_seis_read_hk /NACK_seis_read_hk
seis_read_recentering_data	0x3E	Asks to retrieve the 3 VBB position and velocity values and related temperatures on the low speed interface (last available values)	ACK_seis_read_recentering_data/ NACK_seis_read_recentering_data

Document technique DT6 : Exemple de structure de trame CIP échangée

Protocol layers:

IP v4: Internet Protocol version 4

- ▶ UDP: User Datagram Protocol
- ▶ CIP: Common Industrial Protocol
- ▶ IDP: Internetwork Datagram Protocol

NOOP command sent from SCE:

```

|01|02|03|04 05|06|07|08 09|10|11|12 13|14|15|16 17|18|19|20
0x|45|00|00|28 00|00|40|00 FF|11|F8|70 C0|A8|01|01 C0|A8|01|02
|21|22|23|24 25|26|27|28 29|30|31|32 33|34|35|36 37|38|39|40
0x|13|8A|13|8A 00|14|17|67 EE|FA|4D|EF 00|0C|02|00 FF|00|FF|FF
    
```

Internet Protocol version 4 Header Format : RFC791 (1981) and RFC1349* (1992) *Obsoleted by RFC2474 (1998) and RFC3168 (2001)				
Bit Length	Bit Offset	Definition	Translation	
16 (2 bytes)	0	<ul style="list-style-type: none"> • Version • IHL : Internet Header Length 	$0x 4500 = 0b 0100 0101 0000 0000 $ $\text{-----}0b 0100 = 4 : \text{Version IPv4}$ $\text{-----}0b 0101 = 5 : 5 \times 32 \text{ bit}$ words = 20 bytes	
4	0	<ul style="list-style-type: none"> • TOS : Type Of Service ▶ Precedence 	$0b 0000 0000 $ $0b 111. : \text{Network Control}$ $0b 110. : \text{Internetwork Control}$ $0b 101. : \text{CRITIC/ECP}$ $0b 100. : \text{Flash Override}$ $0b 011. : \text{Flash}$ $0b 010. : \text{Immediate}$ $0b 001. : \text{Priority}$ $0b 000. : \text{Routine}$ $0b. 1000. : \text{Minimize Delay}$ $0b.0 100. : \text{Maximize Throughput}$ $0b.0 010. : \text{Maximize Reliability}$ $0b.0 001. : \text{Minimize Monetary Cost}$ $0b.0 000. : \text{Normal Service}$ $0b.0 : (\text{unused bit})$	
4	8 (1 byte)			▶ TOS Field
1	8			▶ MBZ : Must Be Zero
16 (2 bytes)	16 (2 B)	• Total Length	$0x 0028 = 40 \text{ bytes Complete Frame}$	
Bits	Bit offset	Definition	Translation	
16 (2 bytes)	32 (4 B)	• Identification	$0x 0000 = 0b 0000 0000 0000 0000 :$ Assigned by the sender to aid in assembling the fragments of a datagram.	
16 (2 bytes)	48 (6 B)	<ul style="list-style-type: none"> • Flags ▶ MBZ : Must Be Zero ▶ DF : Don't Fragment 	$0x 4000 = 0b 0100 0000 0000 0000 $ $\text{-----}0b 010. = \text{Don't Fragment}$ $0b 0.. : (\text{unused bit})$ $0b .0.. : \text{May Fragment}$ $0b .1.. : \text{Don't Fragment}$ $0b ..0. : \text{Last Fragment}$ $0b ..1. : \text{More Fragments}$	

13	51	<ul style="list-style-type: none"> ▶ MF : More Fragments • Fragment Offset 	<p>0b ...0 0000 0000 0000 : Indicates where in the datagram this fragment belongs</p>
8 (1 byte)	64 (8 B)	<ul style="list-style-type: none"> • TTL : Time to Live 	<p>Maximum time the datagram is allowed to remain in the internet system. This field is modified in internet header processing. The time is measured in units of seconds, but since every module that processes a datagram must decrease the TTL by at least one even if it process the datagram in less than a second, the TTL must be thought of only as an upper bound on the time a datagram may exist.</p> <p>0x FF = 255 : 256 allowed times</p> <p>NB : When the TTL field hits zero, the router discards the packet and typically sends an ICMP Time Exceeded message to the sender. The program traceroute uses these ICMP Time Exceeded messages to print the routers used by packets to go from the source to the destination.</p>
8 (1 byte)	80 (9 B)	<ul style="list-style-type: none"> • Protocol 	<p>0x 11 = 0b 0001 0001 = 17 : UDP</p> <p>1 = ICMP : Internet Control Message Protocol 2 = IGMP : Internet Group Management Protocol 6 = TCP : Transmission Control Protocol 17 = UDP : User Datagram Protocol</p>
Bits	Bit offset	Definition	Translation
16 (2 bytes)	88 (10 B)	<ul style="list-style-type: none"> • Header Checksum 	<p>The checksum field is the 16 bit one's complement of the one's complement sum of all 16 bit words in the header. For purposes of computing the checksum, the value of the checksum field is zero. Since some header fields change (e.g., time to live), this is recomputed and verified at each point that the internet header is processed.</p> <p><u>Computing the Internet Checksum :</u></p> <p>All the IP Header except Checksum itself : (IP_VERSION << 12) + (IP_HEADER_LEN << 8) + IP_TOS = 0x4500 IP_Length = 0x0028 IP_ID = 0x0000</p>

			<pre> IP_FLAGS << 13 + IP_FRAG_OFFSET = 0x4000 IP_TTL << 8 + IP_PROTOCOL = 0xFF11 IP_Source >> 16 = 0xC0A8 IP_Source & 0xFFFF = 0x0101 IP_Destination >> 16 = 0xC0A8 IP_Destination & 0xFFFF = 0x0102 ctIPHeader = [0x4500,0x0028,0x0000,0x4000,0xFF11,0xC0 A8,0x0101,0xC0A8,0x0102] IP_Checksum = CalculChecksum { wSum = sum(ctIPHeader) while (wSum > 0xFFFF): wSum = (wSum >> 16) + (wSum & 0xFFFF) if (wSum == 0xFFFF): return 0xFFFF return (~wSum) & 0xFFFF } IP_Checksum = 0xF870 </pre>
32 (4 bytes)	104 (12 B)	• Source Address	0x C0 A8 01 01 = "192.168.1.1"
32 (4 bytes)	108 (16 B)	• Destination Address	0x C0 A8 01 02 = "192.168.1.2"

User Datagram Protocol Header Format : RFC768 (1980)			
Bits	Bit offset	Definition	Translation
16 (2 bytes)	116 (20 B)	• Source Port	0x 138A = 5002
16 (2 bytes)	132 (22 B)	• Destination Port	0x 138A = 5002
16 (2 bytes)	148 (24 B)	• Total Length	0x 0014 = 20 bytes Complete UDP Frame with Data
16 (2 bytes)	164 (26 B)	• Checksum	<p>Checksum is the 16-bit one's complement of the one's complement sum of a pseudo header of information from the IP header, the UDP header, and the data, padded with zero octets at the end (if necessary) to make a multiple of two octets.</p> <p><u>Computing the UDP Checksum :</u></p> <p>IP Source IP Destination IP PROTOCOL</p>

			<pre> UDP Length UDP Source UDP Destination UDP Length All CIP headers All IDP data portions UDP_Checksum = CalculChecksum { wSum = sum(ctxxxx) while (wSum > 0xFFFF): wSum = (wSum >> 16) + (wSum & 0xFFFF) if (wSum == 0xFFFF): return 0xFFFF return (~wSum) & 0xFFFF } UDP_Checksum = 0x1767 </pre>
--	--	--	--

Document technique DT7 : Extrait Code socket en C

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h> /* Ce fichier definit sockaddr_in comme : */
                        /* struct sockaddr_in { */
                        /*     short sin_family;  famille d'adresse */
                        /*     u_short sin_port;  numéro de port */
                        /*     struct in_addr sin_addr;  adresse IP */
                        /*     char sin_zero[8]   inutile */

main()
{
    int sock, length;
    struct sockaddr_in name;
    char buf[1024];

    sock = socket(AF_INET,SOCK_DGRAM,0);
    if (sock == -1)
        {
            perror("ouverture socket");
            /*exit(1);*/ return(1);
        }
    name.sin_family = AF_INET;
    name.sin_addr.s_addr = INADDR_ANY;
    name.sin_port = 0;
    if (bind(sock,(struct sockaddr *)&name, sizeof name) == -1)
        {
            perror ("pas de creation");
            /*exit(1);*/ return(1);
        }
    length = sizeof(name);
    if (getsockname(sock, (struct sockaddr *) &name, &length) == -1)
        {
            perror ("pas de numero de port");
            /*exit(1);*/ return(1);
        }
    printf("numero du port #%d\n", ntohs (name.sin_port));
    if (read(sock, buf, 1024) == -1) perror("data");
    printf("-->%s\n", buf);
    close(sock);
    /*exit(0); */ return(0);
}
```

Document technique DT8 : Code de Hamming

Un code de Hamming est un code correcteur linéaire. Il permet la détection et la correction automatique d'une erreur si elle ne porte que sur l'altération d'un caractère du message.

L'objectif de ce code est la détection et la correction d'une erreur après transmission d'un message. Cette correction est permise grâce à l'ajout d'informations redondantes. Le message est envoyé dans un message plus grand, la différence de taille contient la redondance. En cas d'altération du message, la redondance est conçue pour détecter et corriger l'erreur.

Structure du message :

- message de m bits auquel on ajoute les n bits de contrôle de parité C_x (bits de redondance ou de contrôle de réception).
- longueur totale du message en bits, redondance comprise : $2^n - 1$
- longueur totale du message : $m = (2^n - 1) - n$
- place des bits de parité (redondance) aux puissances de 2 :

Place des bits de parité ou redondance								2^3				2^2		2^1	2^0
								8				4		2	1
Position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Bits C_i et m_x	m10	m9	m8	m7	m6	m5	m4	C3	m3	m2	m1	C2	m0	C1	C0
Exemple message 1 Code (7,4)	1	1	0	1	1	0	0	0	1	0	0	0	0	0	0

Après calcul de la parité à la réception, les bits de contrôle prennent la place des bits de redondance

Dans le cas ci-dessus, les bits de contrôle de réception C3C2C1C0 valent 0000, il n'y a pas d'erreur sinon la valeur des bits de contrôle indiquerait la position de l'erreur entre 1 et 2^n-1

Donc, la longueur totale du message 1 est 11bits de message (11011001000) + 4 bits parité et pas d'erreur détectée et à corriger.

Position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Bits C_i et m_x	m10	m9	m8	m7	m6	m5	m4	C3	m3	m2	m1	C2	m0	C1	C0
Exemple message 2 code 15,5	1	1	1	1	1	1	0	1	1	1	0	0	0	1	0

Dans ce cas, les bits de contrôle de réception C3C2C1C0 valent 1010, il y a une erreur qui indique la position de l'erreur entre 1 et 2^4-1

Donc, la longueur totale : 11bits de message (11111101100) + 4 bits parité et une erreur détectée et à corriger à la position 5.

Soit le message corrigé : 11111101110

En complément dans le spatial : Correction d'erreur unique et détection d'erreur double
Un bit de vérification supplémentaire étend également la fonction pour détecter les erreurs de double bit en ajoutant un contrôle qui couvre chaque bit de données.

Exemple : Si on utilise le code (72,64) Hamming, le mot total est de 72 bits, composée de 64 bits de données et de 8 bits de contrôle. Le nombre de bits de contrôle nécessaires pour corriger une erreur de bit unique dans le mot de 64 bits est de 7, + 1 bit de vérification supplémentaire. Si les 7 bits de contrôle donnent une valeur 1 alors que le 8^{ème} bit renvoie un zéro, alors une erreur de double bit a eu lieu.

Si une erreur de double bit se produit, les positions des bits d'erreur ne sont pas identifiées ou mal identifiées.

Document technique DT9 : Code C UART – Fonctions C pour piloter L'UART du FPGA

// Fonction d'initialisation de l'UART – baud est en kilobit par seconde

```
void uart_init(uint32_t baud)
{
    cli();
    UBRR1 = (F_CPU / 4 / baud - 1) / 2;
    UCSR1A = (1<<U2X1);
    UCSR1B = (1<<RXEN1) | (1<<TXEN1) | (1<<RXCIE1);
    UCSR1C = (1<<UCSZ11) | (1<<UCSZ10);
    tx_buffer_head = tx_buffer_tail = 0;
    rx_buffer_head = rx_buffer_tail = 0;
    sei();
}
```

// Transmission d'un caractère c

```
void uart_putchar(uint8_t c)
{
    uint8_t i;

    i = tx_buffer_head + 1;
    if (i >= TX_BUFFER_SIZE) i = 0;
    while (tx_buffer_tail == i);
    tx_buffer[i] = c;
    tx_buffer_head = i;
    UCSR1B = (1<<RXEN1) | (1<<TXEN1) | (1<<RXCIE1) | (1<<UDRIE1);
}
```

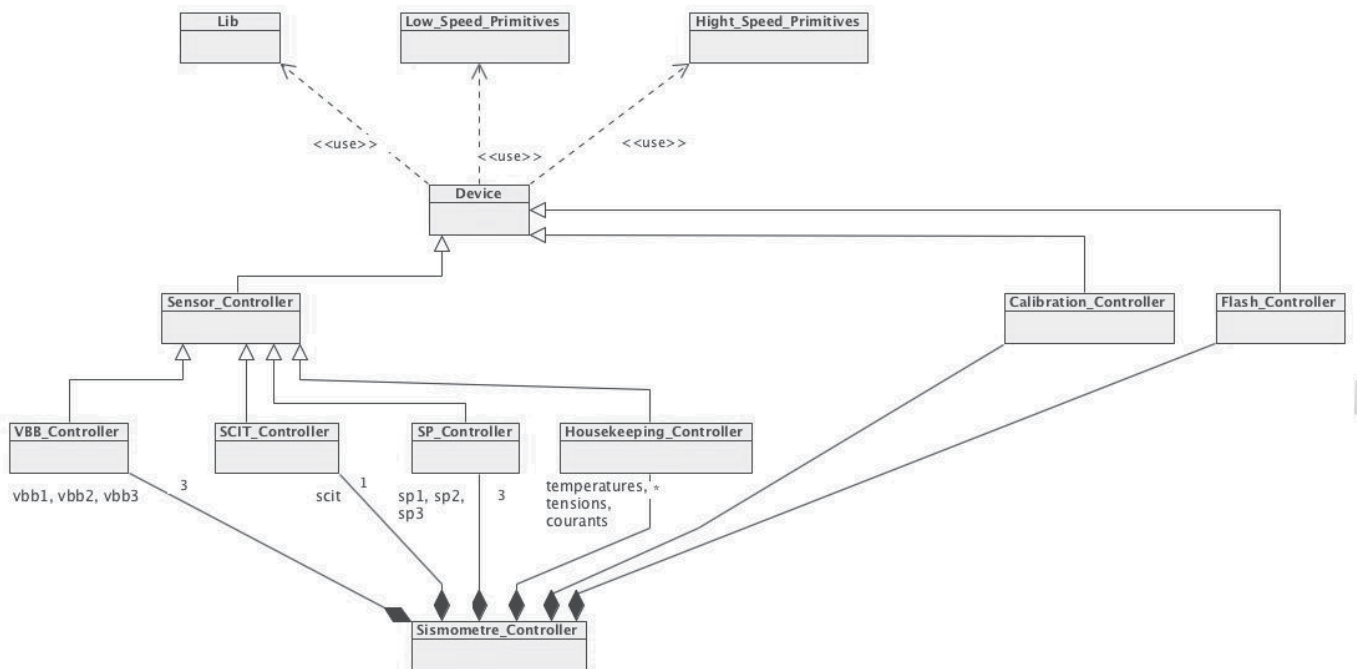
// Réception d'un caractère

```
uint8_t uart_getchar(void)
{
    uint8_t c, i;

    while (rx_buffer_head == rx_buffer_tail);
    i = rx_buffer_tail + 1;
    if (i >= RX_BUFFER_SIZE) i = 0;
    c = rx_buffer[i];
    rx_buffer_tail = i;
    return c;
}
```


NE RIEN ECRIRE DANS CE CADRE

Document technique DR1 : Diagramme de classes à compléter



Document technique DR2 : Spectrogramme à compléter

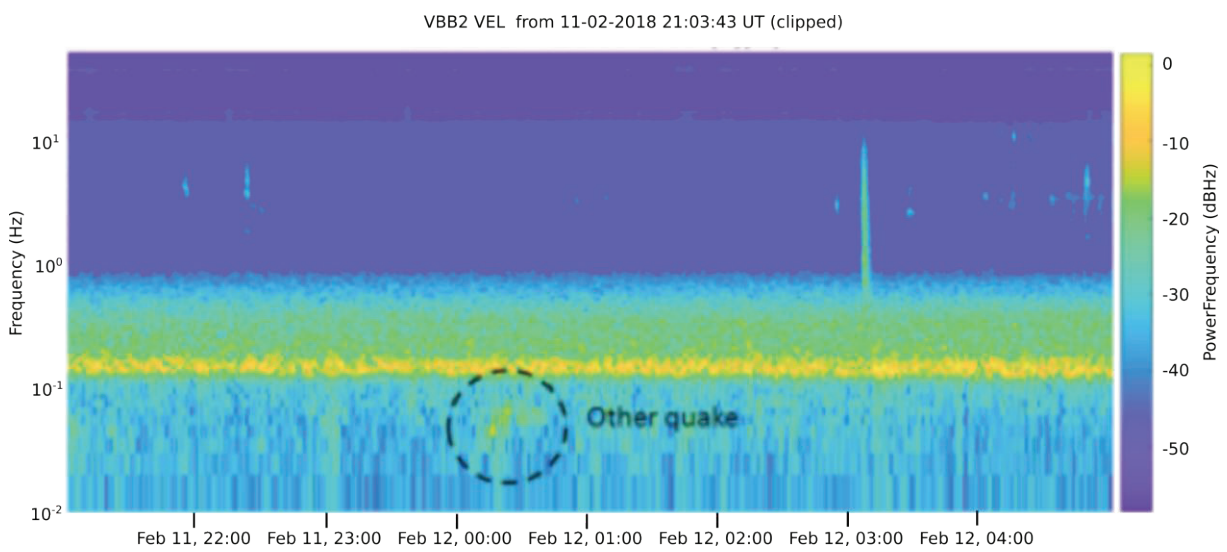


Figure 1 : Diagramme d'observation du séisme de Vendée - données issues du sismomètre (Source CNES)

Document technique DR3 : Code VHDL Mae UART à compléter

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
ENTITY mae IS
PORT(      a : IN   std_logic;
          hor : IN   std_logic;
          raz : IN   std_logic;
          b  : OUT  std_logic );
END mae ;
ARCHITECTURE diagram OF mae IS
TYPE STATE_TYPE IS (Etat0,Etat1,Etat2);
SIGNAL EtatPresent : STATE_TYPE ;
SIGNAL EtatFutur : STATE_TYPE ;
BEGIN
  clocked : PROCESS(hor,raz)
  BEGIN
    IF (raz = '0') THEN
      EtatPresent <= Etat0;
    ELSIF (hor'EVENT AND hor = '1')
      THEN
      EtatPresent <= EtatFutur;
    END IF;
  END PROCESS clocked;
  nextstate : PROCESS (EtatPresent, a)
  BEGIN
    CASE
      EtatPresent IS
        WHEN Etat0 => .....

        .....
        WHEN Etat1 => .....

        .....
        WHEN Etat2 => .....

        .....
        WHEN OTHERS => .....

        .....
    END CASE;
  END PROCESS nextstate;
END ARCHITECTURE diagram ;
```


NE RIEN ECRIRE DANS CE CADRE

Document technique DR4 : Extrait code Python à compléter

Pour information, seules les descriptions des types et variables utiles sont précisées.

Extrait code Python :

```
from ctypes import (
    BigEndianStructure,
    c_int16,
    c_ubyte,
    c_uint16,
)
from lib.Constants import *
# from lib.Hex_Tools import *

IP_IHL_VERSION = 0x45
IP_TYPE_OF_SERVICE = 0x00
IP_ID = 0x0000
IP_FRAGMENT_OFFSET_FLAGS = (0x40, 0x00)
IP_TIME_TO_LIVE = 0xFF # 255 in decimal
IP_PROTOCOL = 0x11 # UDP 17 in decimal

IP_ADDRESS_CONTROLLER = (0xC0, 0xA8, 0x01, 0x01) # (192.162.1.1) C&CH
IP_ADDRESS_INSTRUMENT = (0xC0, 0xA8, 0x01, 0x02) # (192.162.1.2)
# MY_IP_ADDRESS = IP_ADDRESS_CONTROLLER

UDP_SRC_PORT = 5002
UDP_DST_PORT = 5002

CIP_MARKER = (0xEE, 0xFA, 0x4D, 0xEF)
CIP_RESERVED = 0x00

LVL_MDE_READ_OP = 0x20
LVL_MDE_WRITE_OP = 0x40

VBB_CALIBRATION_AREA = 0x00
SP_CALIBRATION_AREA = 0x01

MDE_MODE_OFF = 0x00
MDE_MODE_HEATER = 0x01
```

```
MDE_MODE_STANDBY = 0x02
MDE_MODE_MOTOR_ON = 0x03
```

```
# IP Header structure
```

```
class IP_Header_t(BigEndianStructure):
    _pack_ = 1
    _fields_ = [
        ('IHL_Version', c_ubyte),           # 0: 1 byte (4 bits + 4 bits)
        ('TypeOfService', c_ubyte),        # 1: 1 byte
        ('TotalLength', c_int16),          # 2: 2 bytes
        ('ID', c_int16),                   # 4: 2 bytes
        ('FragmentOff_Flags', c_ubyte * 2), # 6: 2 bytes (3 bits + 13 bits)
        ('TimeToLive', c_ubyte),           # 8: 1 byte
        ('Protocol', c_ubyte),              # 9: 1 byte
        ('HeaderChecksum', c_uint16),      # 10: 2 bytes
        ('SourceAddress', c_ubyte * 4),    # 12: 4 bytes
        ('DestinationAddress', c_ubyte * 4) # 16: 4 bytes
    ]
```

```
def __init__(
    self,
    ihl_version=IP_IHL_VERSION,
    typeOfService=IP_TYPE_OF_SERVICE,
    totalLength=0x0000,
    identifier=IP_ID,
    fragmentOff_Flags=IP_FRAGMENT_OFFSET_FLAGS,
    timeToLive=IP_TIME_TO_LIVE,
    protocol=IP_PROTOCOL,
    headerChecksum=0x0000,
    srcAddr=(0x00, 0x00, 0x00, 0x00),
    destAddr=(0x00, 0x00, 0x00, 0x00)
):
    self.IHL_Version = ihl_version
    self.TypeOfService = typeOfService
    self.TotalLength = totalLength
    self.ID = identifier
    self.FragmentOff_Flags = fragmentOff_Flags
    self.TimeToLive = timeToLive
    self.Protocol = protocol
    self.HeaderChecksum = headerChecksum
    self.SourceAddress = srcAddr
    self.DestinationAddress = destAddr
```

```
# Init IP_Header Instrument et Controller à compléter
```

```
IP_header_to_instrument =
    IP_Header_t(..... , .....)

```

```
IP_header_to_controller =
    IP_Header_t(..... , .....)

```

Document technique DR5 : Format du type de trame échangée entre C&DH <-> E-BOX

Formats type

Commande :

	<i>IP Header</i>	<i>UDP header</i>	<i>CIP Header</i>	<i>IDP</i>
	<i>20 octets</i>	<i>8 octets</i>	<i>8 octets</i>	<i>Command ID (1 octet), Data(n octets), Padding</i>

Réponse :

	<i>IP Header</i>	<i>UDP header</i>	<i>CIP Header</i>	<i>IDP</i>
	<i>20 octets</i>	<i>8 octets</i>	<i>8 octets</i>	<i>ACK/NACK (1 octet), Command ID (1 octet), Data(n octets), Padding</i>

A compléter :

Trame de vers

<i>IP Header</i>	<i>UDP header</i>	<i>CIP Header</i>	<i>IDP</i>
<i>20 octets</i>	<i>8 octets</i>	<i>8 octets</i>	<i>À compléter</i>

Trame de vers

<i>IP Header</i>	<i>UDP header</i>	<i>CIP Header</i>	<i>IDP</i>
<i>20 octets</i>	<i>8 octets</i>	<i>8 octets</i>	<i>À compléter</i>

Trame de vers

<i>IP Header</i>	<i>UDP header</i>	<i>CIP Header</i>	<i>IDP</i>
<i>20 octets</i>	<i>8 octets</i>	<i>8 octets</i>	<i>À compléter</i>

Trame de vers

<i>IP Header</i>	<i>UDP header</i>	<i>CIP Header</i>	<i>IDP</i>
<i>20 octets</i>	<i>8 octets</i>	<i>8 octets</i>	<i>À compléter</i>

Nom de famille :

(Suivi, s'il y a lieu, du nom d'usage)



Prénom(s) :

Numéro
Inscription :

Né(e) le :

(Le numéro est celui qui figure sur la convocation ou la feuille d'émargement)

(Remplir cette partie à l'aide de la notice)

Concours / Examen : Section/S spécialité/Série :

Epreuve : Matière : Session :

CONSIGNES

- Remplir soigneusement, sur CHAQUE feuille officielle, la zone d'identification en MAJUSCULES.
- Ne pas signer la composition et ne pas y apporter de signe distinctif pouvant indiquer sa provenance.
- Numéroter chaque PAGE (cadre en bas à droite de la page) et placer les feuilles dans le bon sens et dans l'ordre.
- Rédiger avec un stylo à encre foncée (bleue ou noire) et ne pas utiliser de stylo plume à encre claire.
- N'effectuer aucun collage ou découpage de sujets ou de feuille officielle. Ne joindre aucun brouillon.

EAE SIN 3

DR6

Tous les documents réponses sont à rendre, même non complétés.

NE RIEN ECRIRE DANS CE CADRE

Document technique DR6 : Chronogramme à compléter

