

SESSION 2026



CAPET
CONCOURS EXTERNE ET CAFEP CORRESPONDANT
(BAC + 3)

Section : SCIENCES INDUSTRIELLES DE L'INGÉNIEUR

Option : INGÉNIERIE INFORMATIQUE

ADMISSIBILITÉ : ÉPREUVE 1

L'épreuve consiste en l'étude de la modélisation d'un produit.

Elle a pour objectif de vérifier la capacité du candidat à conduire une étude d'un produit pour en caractériser les performances en mobilisant ses connaissances scientifiques et technologiques relevant de l'option du concours.

L'épreuve est constituée de plusieurs parties indépendantes, certaines d'entre elles sont à traiter obligatoirement et d'autres sont au choix du candidat.

Durée : 4 heures

Calculatrice autorisée selon les modalités de la circulaire du 17 juin 2021 publiée au BOEN du 29 juillet 2021.

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

Il appartient au candidat de vérifier qu'il a reçu un sujet complet et correspondant à l'épreuve à laquelle il se présente.

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.

Le fait de rendre une copie blanche est éliminatoire.

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 externe du CAPET de l'enseignement public :**

Concours	Section/option	Epreuve	Matière
LDE	1413E	101	4061

► **Concours externe du CAFEP/CAPET de l'enseignement privé :**

Concours	Section/option	Epreuve	Matière
LDF	1413E	101	4061

Définition de l'épreuve

L'épreuve consiste en l'étude de la modélisation d'un produit.

Elle a pour objectif de vérifier la capacité du candidat à conduire une étude d'un produit pour en caractériser les performances en mobilisant ses connaissances scientifiques et technologiques relevant de l'option du concours.

L'épreuve est constituée de plusieurs parties indépendantes, certaines d'entre elles sont à traiter obligatoirement et d'autres sont au choix du candidat.

Conseils aux candidats

Il est demandé aux candidats :

- de rédiger les réponses aux différentes parties sur des feuilles de copie séparées et clairement repérées ;
- de numéroter chaque feuille de copie et indiquer le numéro de la question traitée ;
- de rendre tous les documents réponses, même non complétés ;
- d'utiliser exclusivement les notations indiquées dans le sujet lors de la rédaction des réponses ;
- de justifier clairement les réponses ;
- d'encadrer ou souligner les résultats ;
- de présenter lisiblement les applications numériques, sans omettre les unités, après avoir explicité les expressions littérales des calculs ;
- de formuler les hypothèses nécessaires à la résolution des problèmes posés si celles-ci ne sont pas indiquées dans le sujet.

Sommaire

	Page
Sommaire	1
Présentation du support	2
Partie A (à traiter obligatoirement)	3
Partie B (à traiter obligatoirement)	6
Partie C (à traiter obligatoirement)	9
Partie D (Choix 1)	11
Partie E (Choix 2)	14
DOCUMENTS RÉPONSES	17
DOCUMENTS TECHNIQUES	24

Les parties A, B et C sont à traiter obligatoirement par tous les candidats.

Les candidats devront choisir de traiter la partie D (choix 1) ou la partie E (choix 2).

Présentation du support

Système de supervision d'un microgrid

Dans le cadre de ce sujet, l'étude porte sur la supervision d'un microgrid (réseau électrique localisé) installé dans une entreprise du domaine des énergies renouvelables, qui a pour objectif de rendre son siège français quasiment autonome en électricité.

D'un point de vue puissance, le microgrid étudié est constitué :

- de 230 m² de panneaux voltaïques sur les toits de la société pour une puissance crête de 40 kW.
- de 8 bornes de recharge de véhicules électriques gérées par un système de gestion de recharge EVLink LMS de chez Schneider.
- de batteries de stockage d'une capacité totale de 40 kW·h, permettant de stocker l'énergie électrique lorsque la production photovoltaïque est excédentaire.

Afin de contrôler les flux de puissances et s'assurer du bon fonctionnement des installations, la société Naiobee a mis en place un système de supervision du microgrid, articulé autour d'un composant central : la Naiobox.

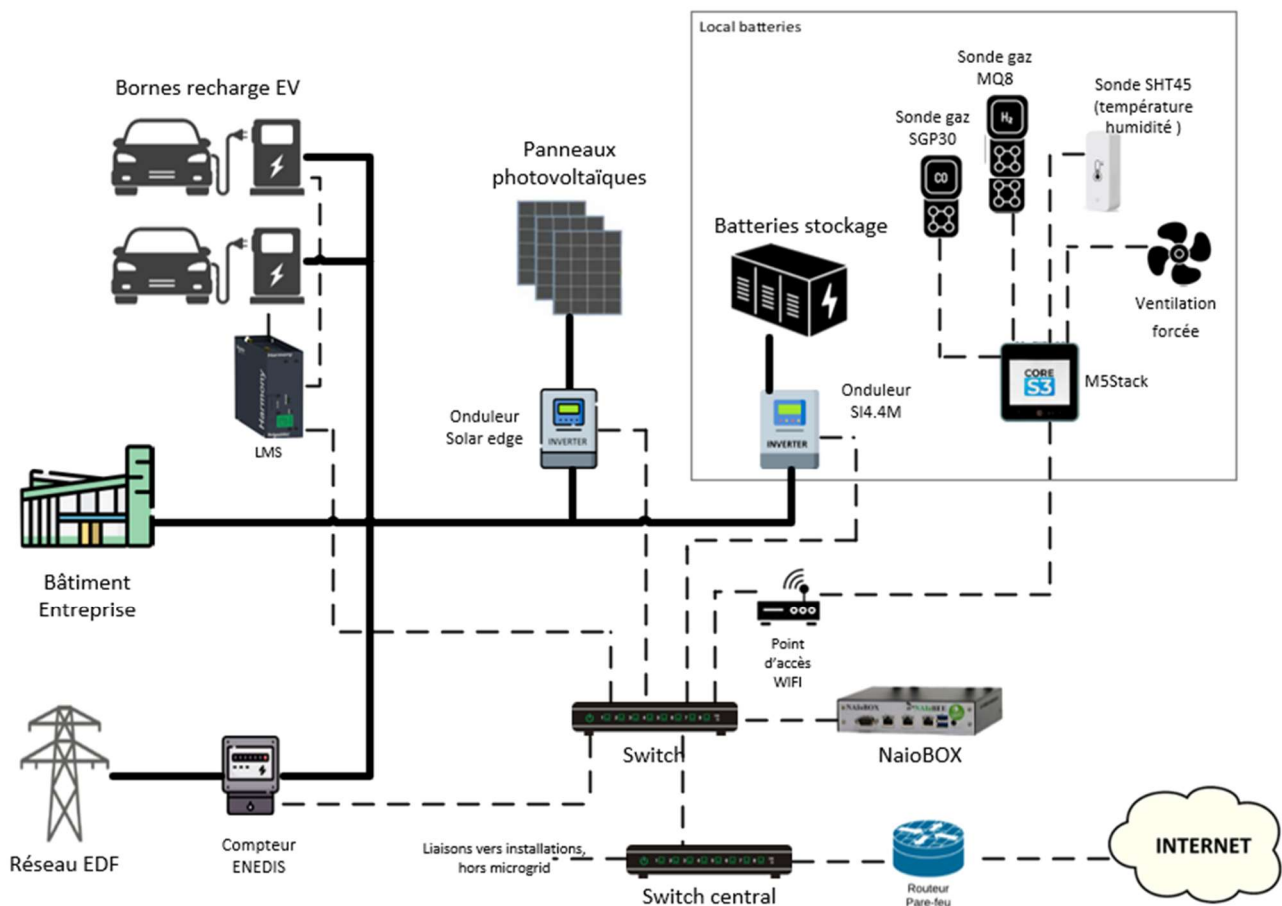


Figure 1 : Microgrid supervisé par Naiobox

Partie A (à traiter obligatoirement)

Mesure du taux de dihydrogène par capteur analogique

En cas de défaillance de la batterie de stockage de l'énergie, celle-ci peut entraîner un dégagement de différents gaz, dont du dihydrogène H₂. On envisage donc d'installer par sécurité un capteur de référence MQ8 dans le local batterie, afin de mesurer le taux de dihydrogène. Ce capteur délivre un signal analogique V_{RL} qui va être renvoyé sur un port GPIO d'une carte à microcontrôleur M5Stack basé sur l'ESP32.

Objectif : Mettre en œuvre le capteur analogique MQ8 et programmer la mesure de taux de H₂ en python.

On donne, en figure A1, les principales fonctionnalités disponibles sur les ports du M5Stack :

GPIO TYPE	Analog Function	M-BUS		Analog Function	GPIO TYPE		
		LINE 0	LINE 1				
		GND	ADC	G35	ADC1_CH7	I	
		GND	ADC	G36	ADC1_CH0	I	
		GND	RST	EN			
I/O/T		G23	MOSI	DAC/SPK	G25	ADC2_CH8	I/O/T
I/O/T		G19	MISO	DAC	G26	ADC2_CH9	I/O/T
I/O/T		G18	SCK	3.3V			
I/O/T		G3	RXD1	TXD1	G1		I/O/T
I/O/T		G16	RXD2	TXD2	G17		I/O/T
I/O/T		G21	SDA	SCL	G22		I/O/T
I/O/T	ADC2_CH2/T2	G2	GPIO	GPIO	G5		I/O/T
I/O/T	ADC2_CH5	G12	IIS_SK	IIS_WS	G13	ADC2_CH4/T4	I/O/T
I/O/T	ADC2_CH3/T3	G15	IIS_OUT	IIS_MK	G0	ADC2_CH1/T1	I/O/T
		HPWR	IIS_IN	G34	ADC1_CH6	I	

Figure A1 : Ports M5Stack

Question A1 | Proposer et justifier le choix d'un numéro de port du M5Stack pouvant être utilisé pour recueillir le signal issu du capteur MQ8.

Le capteur MQ8 est alimenté sous 5 V alors que le microcontrôleur ESP32 du M5Stack fonctionne sous une tension de 3,3 V. On doit donc insérer un montage permettant d'adapter le niveau de tension. On propose, ci-dessous, un circuit dans lequel on a choisi R₂ = 10 kΩ.

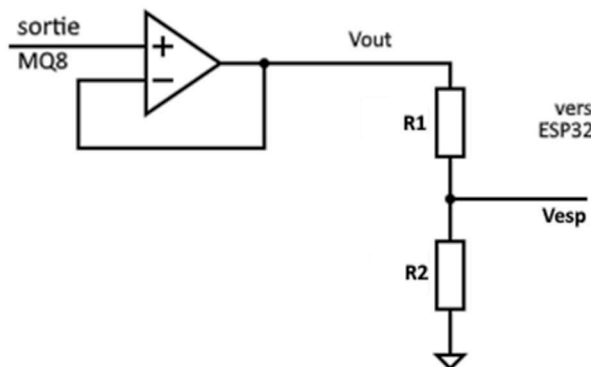


Figure A2 : Circuit d'adaptation de tension pour ESP32

Question A2 | Calculer la valeur à imposer à R₁ de manière à ce qu'une tension V_{out} de 5V entraîne une tension V_{esp} de 3,3 V sur le port de l'ESP32.

On donne ci-dessous le schéma permettant la mise en œuvre du capteur MQ8. La tension V_{RL} constitue la grandeur de sortie du capteur. Cette tension dépend de la résistance de charge R_L à placer en série avec la partie active du capteur.

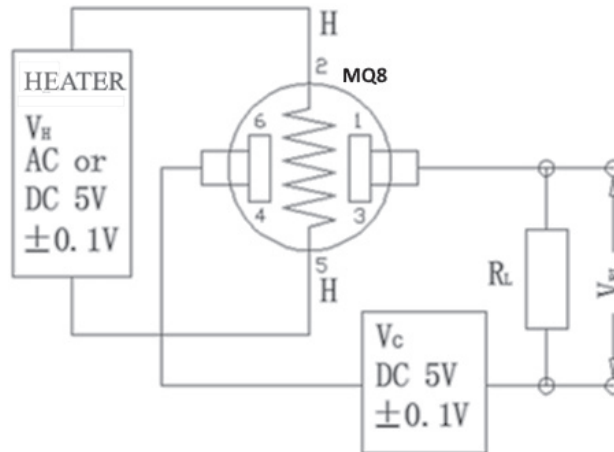


Figure A3 : Circuit interne MQ8

Question A3 | Donner le nom du montage entre la sortie du capteur MQ8 et V_{out} (figure A2). Indiquer son intérêt dans ce cas de figure.

Le convertisseur analogique numérique (CAN) intégré dans l'ESP32 du M5Stack a une résolution de 12 bits et travaille sur la plage de tension d'entrée [0 ; 3,3 V].

Question A4 | Indiquer ce que représente la résolution d'un CAN. En déduire la valeur du quantum du CAN (la plus petite variation de tension détectable par le CAN ou tension correspondant à un incrément de 1 en sortie de CAN).

Dans les conditions d'utilisation du capteur, on a relevé, ci-dessous, la relation entre tension au niveau du CAN de l'ESP32 et concentration en dihydrogène en ppm (parties par million).

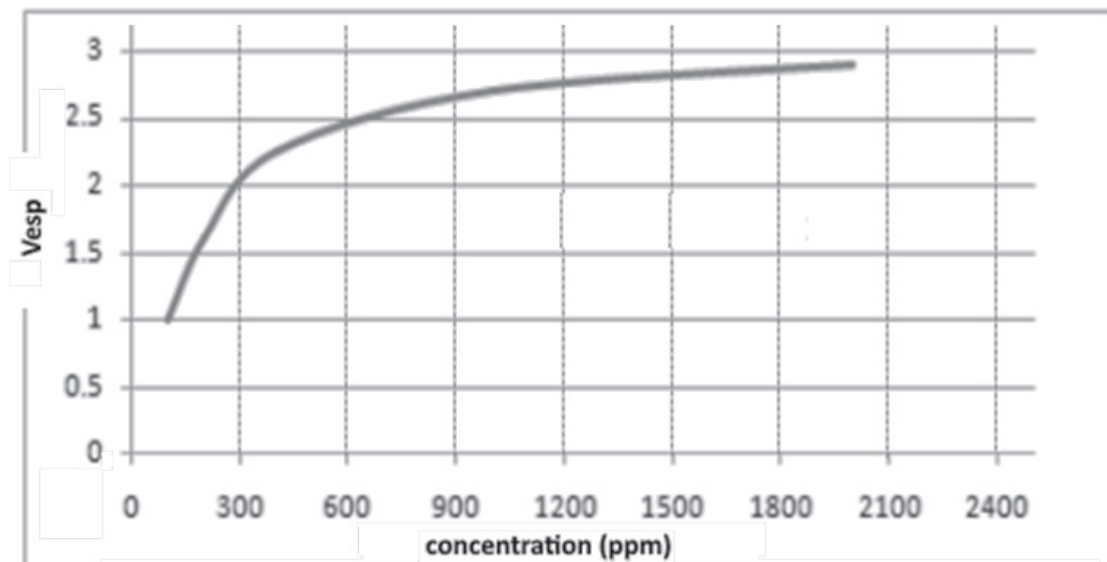


Figure A4 : Tension CAN ESP32 en fonction de la concentration en H2

Question A5 | Calculer le nombre N_{seuil} qui sera délivré par le CAN de l'ESP32 pour une concentration d'hydrogène de 300 ppm, seuil à partir duquel une alerte sera déclenchée et la ventilation mise en route. Donner sa représentation en binaire.

Tournez la page S.V.P.

Dans les questions suivantes, l'objectif est de coder en Python les fonctions qui permettront d'obtenir la valeur du taux de concentration en dihydrogène à partir de la valeur de sortie du CAN.

Question A6 | Donner le code de la fonction `convCAN` qui prend l'argument `valeurCAN`, un entier correspondant à la valeur de sortie du CAN et renvoie la valeur de la tension appliquée sur le CAN correspondante.

Afin d'obtenir par programmation la correspondance entre la tension délivrée par le capteur MQ8 et la concentration en dihydrogène, on dispose d'une table représentant la courbe donnée en figure A4. On donne, ci-dessous, un extrait partiel de la déclaration de cette table simplifiée, sous la variable locale `corresMQ8`.

`corresMQ8 = [(1.0, 103),(1.2, 132),(1.4, 163),(1.6, 197),(1.8, 243),(2.0, 281), ...]`

Question A7 | Donner la fonction `convTauxH2` qui, pour une valeur de tension `Vesp` envoyée par le capteur MQ8, en argument de la fonction, renvoie le taux de dihydrogène correspondant.
La fonction devra renvoyer le taux de H₂ correspondant à la dernière tension de la table inférieure à `Vesp` si la valeur de la tension est bien comprise entre les valeurs extrêmes de la table.
Par exemple, avec la table `corresMQ8`, pour une tension `Vesp` de 1,5 V, la fonction doit renvoyer un taux de 163 ppm.

En fait, le CAN intégré dans l'ESP 32 comporte des non-linéarités entre la tension d'entrée et la valeur renvoyée par le CAN. On donne, figure A5, la relation entre tension et valeur renvoyée :

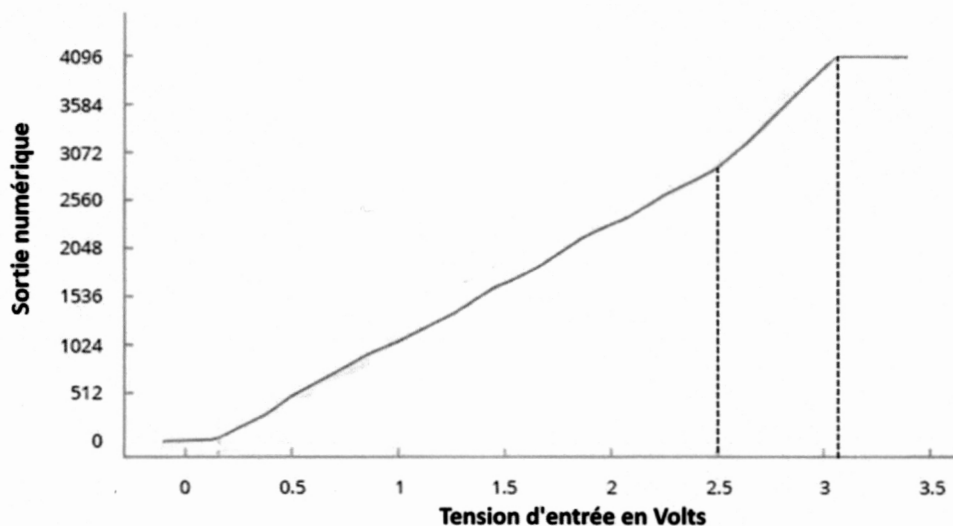


Figure A5 : Caractéristique CAN ESP32

Question A8 | Conclure alors sur la capacité du dispositif et des fonctions étudiés précédemment à fournir une valeur correcte du taux de dihydrogène. Indiquer si la détection de seuil à 300 ppm est réalisable dans ces conditions.

Partie B (à traiter obligatoirement)

Mesure du taux de composants volatiles COVT

Une mesure complémentaire, moins spécifique que la mesure du taux de H₂ vue précédemment, pour mesurer d'éventuels dégagements gazeux dus à un dysfonctionnement des batteries, est d'utiliser un capteur mesurant le taux de composés organiques volatils totaux (COVT). On se propose d'étudier l'utilisation du capteur SGP30 afin de mesurer le taux de COVT.

Objectif : Programmer l'ESP32 du M5stack (programmation objet en C++) afin de mesurer le taux de COVT et déclencher la ventilation en cas de dégagement gazeux anormal.

On souhaite effectuer une mesure toutes les 10 secondes et on définit initialement 2 seuils pour la surveillance du taux de COVT :

- Message d'alerte si le taux de COVT est supérieur à 500 parties par milliard (ppb) ou si la croissance de ce taux est supérieure à 200 ppb par minute.
- Ventilation si le taux de COVT est supérieur à 1000 ppb.

Les caractéristiques générales du SGP30 sont données dans le tableau de la figure B1.

Parameter	Signal	Values		Comments
Output range	TVOC signal	0 ppb to 60000 ppb		Maximum possible output range. The gas sensing performance is specified for the measurement range as defined in Table 1
	CO ₂ eq signal	400 ppm to 60000 ppm		
		Range	Resolution	
	TVOC signal	0 ppb - 2008 ppb	1 ppb	
		2008 ppb - 11110 ppb	6 ppb	
		11110 ppb - 60000 ppb	32 ppb	
	CO ₂ eq signal	400 ppm - 1479 ppm	1 ppm	
		1479 ppm - 5144 ppm	3 ppm	
		5144 ppm - 17597 ppm	9 ppm	
		17597 ppm - 60000 ppm	31 ppm	
Sampling rate	TVOC signal	1 Hz		The on-chip baseline compensation algorithm has been optimized for this sampling rate. The sensor shows best performance when used with this sampling rate.
	CO ₂ eq signal	1 Hz		

Figure B1 : caractéristiques du capteur SGP30

Question B1

Justifier le choix du capteur SGP30 par rapport aux besoins définis précédemment.

La séquence de communication typique entre le maître I2C (par exemple, un microcontrôleur dans un dispositif hôte) et le capteur est décrite comme suit :

1. Le capteur est mis sous tension et la communication est initialisée. Le maître I2C envoie les commandes **sgp30_get_feature_set** (méthode begin), puis **sgp30_iaq_init**. Le capteur répond à chaque message.
2. Le maître I2C demande périodiquement des mesures et lit les données, selon la séquence suivante :
 - a. Le maître I2C envoie une commande de mesure (**sgp30_measure_iaq**).
 - b. Le capteur renvoie un message contenant les taux de concentration en CO2eq (en ppm) et de COVT (en ppb).
 - c. Le maître I2C lit le résultat de la mesure.

Question B2 | Compléter le diagramme de séquence simplifié fourni en DRB1 qui décrit les échanges entre ESP32 et SGP30

Pour gérer la mesure du taux de COVT, on utilise une bibliothèque dédiée, définissant la classe SGP30. La classe SGP30 est spécifiée dans les fichiers SGP30.h et SGP30.cpp

Question B3 | Indiquer comment sont appelés les fichiers de type .h et quels sont leurs rôles.

La gestion du capteur commence par l'instanciation d'un objet SGP30 , que l'on nomme capteurCovt_Batt, par l'instruction : SGP30 capteurCovt_Batt;

Question B4 | Indiquer comment on appelle la méthode permettant d'instancier un objet d'une classe.

La classe SGP30 utilise les classes I2CDevice et TwoWire pour gérer la liaison I2C. On donne en DTB1 la structure simplifiée de la classe SGP30 dans un diagramme de classe.

Question B5 | Indiquer quel est le type de relation entre les classes TwoWire, I2CDevice et SGP30. Expliquer le lien de dépendance entre l'objet composite (conteneur) et l'objet composant (partie) dans ce cas.

Question B6 | Donner les 2 instructions qui permettent d'initialiser l'objet capteurCovt_Batt avec les paramètres par défaut (méthodes begin et IAQinit) .

Le lancement d'une mesure se fait en appelant la méthode IAQmeasure. Cette méthode stocke la valeur du taux de COVT dans l'attribut TVOC.(La méthode renvoie true si la mesure a réussi, false sinon). On donne en DRB2 le code partiel qui s'exécute en boucle, permettant de relever le taux de COVT.

On souhaite stocker le taux de COVT dans une variable locale nommée tauxTvoc.

Question B7 | Indiquer (et justifier) si l'attribut TVOC est public ou privé. En ligne 12 de DRB2, déclarer et réaliser l'affectation de `tauxTvoc`.

On veut stocker le taux de croissance du taux de COVT en ppb par minute dans la variable `slopeCOVT`.

Question B8 | Lignes 15 à 17 de DRB2, déclarer cette variable et donner les instructions qui permettent de réaliser le calcul.

On considérera que la ventilation forcée est activée par appel de la fonction `triggerVentilation()` et arrêtée par appel de la fonction `stopVentilation()`. Par ailleurs, le message d'alerte est lancé par la fonction `alertCOVT()`.

Question B9 | Compléter, dans DRB2 à partir de la ligne 19, le code permettant de gérer le déclenchement de la ventilation et le lancement du message d'alerte en fonction des indications données en début de cette partie de sujet.

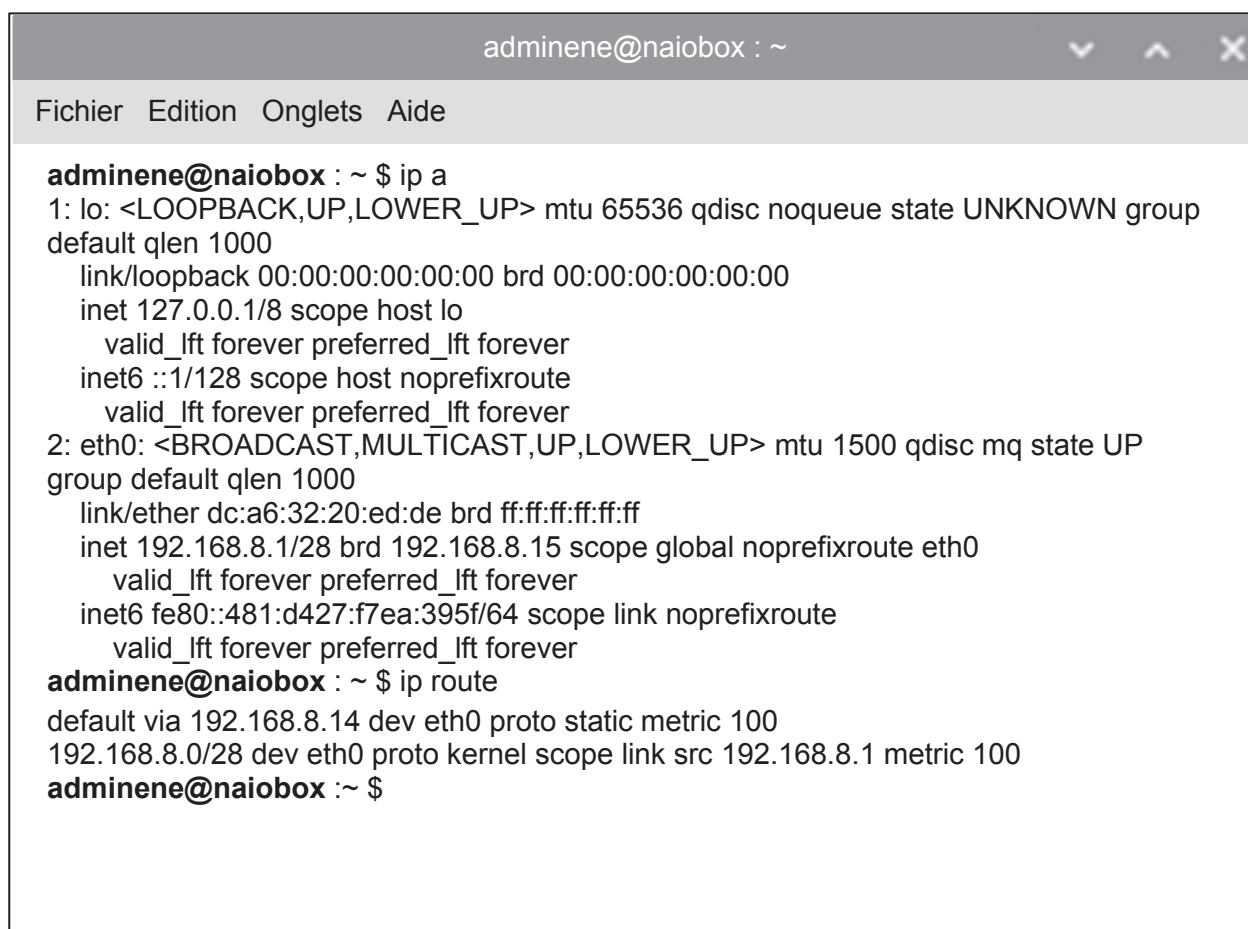
Partie C (à traiter obligatoirement)

Analyse de la communication Naiobox – Onduleur de panneaux solaires

Toutes les 5 secondes, la Naiobox utilise un réseau informatique pour demander aux différents équipements du microgrid (onduleur des panneaux solaires, onduleur de batteries, LMS, compteur ENEDIS...) des informations (tensions, intensités, puissances...) sur l'énergie électrique produite, stockée ou consommée.

Objectif : Analyser la communication entre la Naiobox et l'onduleur des panneaux solaires.

Afin d'identifier les paramètres réseau de la Naiobox, on saisit la commande décrite à la figure C1.



```
adminene@naiobox : ~
Fichier Edition Onglets Aide
adminene@naiobox : ~ $ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP
group default qlen 1000
    link/ether dc:a6:32:20:ed:de brd ff:ff:ff:ff:ff:ff
    inet 192.168.8.1/28 brd 192.168.8.15 scope global noprefixroute eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::481:d427:f7ea:395f/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
adminene@naiobox : ~ $ ip route
default via 192.168.8.14 dev eth0 proto static metric 100
192.168.8.0/28 dev eth0 proto kernel scope link src 192.168.8.1 metric 100
adminene@naiobox : ~ $
```

Figure C1 : Console de la Naiobox

Question C1

Identifier la configuration réseau de la Naiobox puis compléter le tableau du document réponse DRC1. À ce stade, toutes les cases de la colonne Naiobox ne peuvent être complétées.

On poursuit l'identification des composants en utilisant l'analyseur de paquets Wireshark afin de capturer les communications entre la Naiobox et l'onduleur. Les résultats sont présentés dans les documents techniques DTC1, DTC2, DTC3.

- Question C2** | Compléter le tableau du document réponse DRC1 à l'aide du document technique DTC1.
- Question C3** | Indiquer quels filtres ont été appliqués à la trace Wireshark pour obtenir la copie d'écran de documents techniques DTC1. La ligne de commande au format Wireshark n'est pas demandée.
- Question C4** | À l'aide du document technique DTC1, compléter le schéma du document réponse DRC2. Citer les couches du modèle OSI, du modèle TCP/IP et les protocoles utilisés dans la communication Naiobox / onduleur des panneaux solaires pour chacune des couches.
- Question C5** | Modbus est à l'origine une communication série (Modbus RTU). Ici, dans le protocole Modbus TCP, les données Modbus sont encapsulées dans une trame TCP. Expliquer ce que cela signifie. Citer deux avantages à cette solution par rapport au Modbus série.
- Question C6** | Expliquer à quoi servent les trames 85 à 87 (documents techniques DTC1).
- Question C7** | À l'aide du document technique DTC4, indiquer l'utilité du flag PSH (push) présent dans le document technique DTC2 pour l'expéditeur puis pour le récepteur.
- Question C8** | À partir des documents techniques DTC1, DTC4, indiquer quelles sont les données demandées par le client dans la trame 152.
- Question C9** | À partir des documents techniques DTC3, DTC4, indiquer quelles sont l'information du microgrid et sa valeur qui sont transmises à la Naiobox pour être stockées dans la base de données. Justifier votre réponse.
- Question C10** | Indiquer à quoi servent les trames 166 et 167 du document technique DTC1.
- Question C11** | Compléter le diagramme de séquence du document réponse DRC3 afin qu'il modélise le transfert de données entre la Naiobox et l'onduleur pour les traces Wireshark enregistrées.

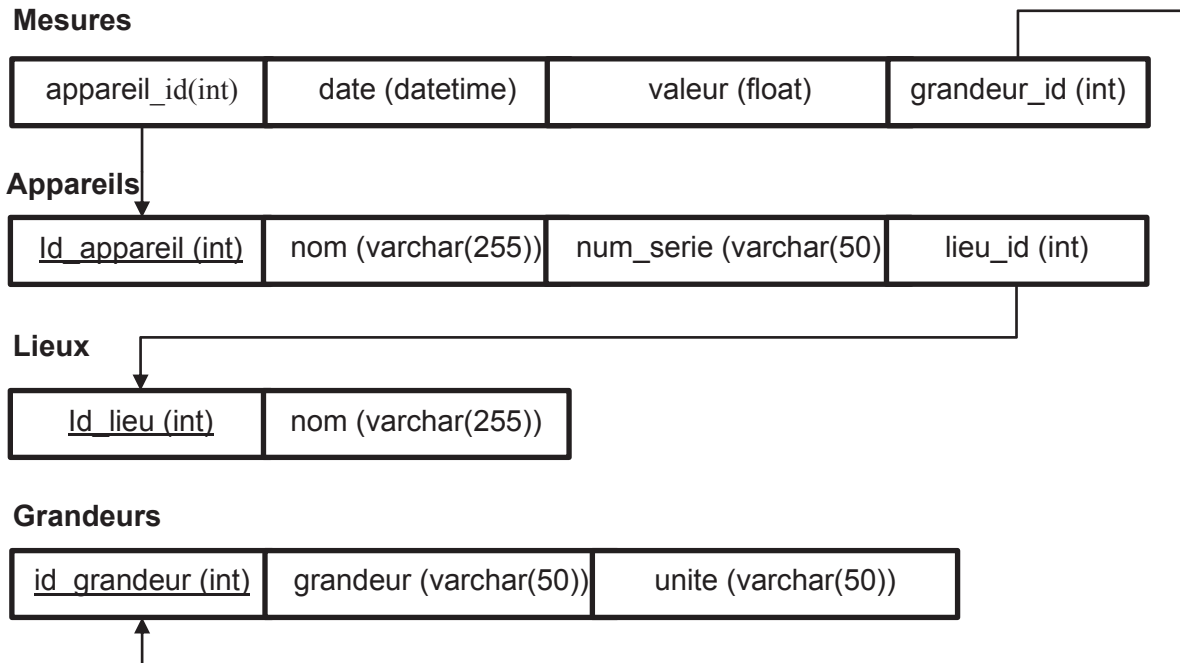
Partie D (Choix 1)

Conception des requêtes SQL

Une base de données SQLite est hébergée dans la Naiobox. Plusieurs flux Node Red sont exécutés sur la Naiobox pour collecter, formater puis enregistrer les données dans la base de données.

Objectif : Écrire une partie des requêtes SQL du système de gestion de bases de données.

La base de données est organisée suivant le schéma relationnel suivant :



Les types des différents attributs des tables **Mesures**, **Appareils**, **Lieux** et **Grandeurs** sont indiqués entre parenthèses. Les clés primaires de chacune des tables sont soulignées. On note les contraintes suivantes dans les données de la table :

- La table **Mesures** contient l'identifiant unique de l'appareil dont provient la mesure, la date et l'heure de la mesure, la valeur mesurée ainsi que l'identifiant de la grandeur physique mesurée.
- La table **Appareils** décrit chaque appareil du microgrid relié au réseau (onduleurs de panneaux solaires, onduleurs de batteries...) en spécifiant son index (`Id_appareil`) son nom, son numéro de série et un identifiant unique du lieu où se trouve l'équipement.
- La table **Lieux** répertorie le lieu où peuvent être placés les équipements ainsi qu'un identifiant unique.
- La table **Grandeurs** répertorie les grandeurs physiques qui peuvent être mesurées en indiquant un identifiant unique, le nom de la grandeur physique mesurée et son unité.

La base de données SQLite gère les dates au format DATETIME qui est un Texte (iso-8601). Les dates contiennent aussi l'heure sous la forme : AAAA-MM-JJ HH:MM:SS).

Exemples :

- Le 1^{er} janvier 2025 à 11h00 s'écrit : '2025-01-01 11:00:00'
- Le 14 juillet 2025 à 23h55 et 10 secondes s'écrit '2025-07-14 23:55:10'

Un rappel des mots clés et fonctions SQL usuels est fourni dans le document technique DTD1.

Question D1 | Donner les requêtes SQL qui ont permis la création de la table **Lieux** suivante avec ses valeurs.

Id_lieu	nom
5	Toiture
4	Local Technique 3er étage
3	Local Technique 2er étage
2	Local Technique 1er étage
1	Local Technique rez-de-chaussée

Question D2 | Indiquer les avantages de stocker le nom dans une table à part et de faire une jointure avec la table **Appareils** plutôt que d'écrire directement le nom du lieu dans la table **Appareils**.

Question D3 | Créer la requête SQL permettant d'ajouter une mesure de 11 245 W effectuée le 25/08/2025 à 12h00 par l'onduleur batterie (id_appareil = 2) dans la table **Mesures**. Pour cette requête, grandeur_id=5.

Question D4 | Proposer une requête de sélection permettant de vérifier que l'ajout a bien été réalisé.

Question D5 | Créer une requête qui liste les appareils situés au "Local Technique rez-de-chaussée", avec leur nom et leur numéro de série.

Question D6 | Créer une requête SQL permettant de rechercher toutes les mesures de puissance AC de l'onduleur de panneaux solaires (id_appareil = 3) pour la date du 7 septembre 2025. Les résultats devront être classés du plus ancien au plus récent et devront fournir l'id_mesure, la date et l'heure de mesure, la valeur de la puissance AC, la grandeur physique mesurée (grandeur_id=5) et l'unité de chaque enregistrement.

Question D7 | Créer une requête qui calcule l'énergie totale produite par les panneaux solaires sur la journée du 7/9/2025 en kilowatt heure (kW·h). L'onduleur de panneaux solaires a pour index : (appareil_id=3) et la mesure a pour index de grandeur (grandeur_id=5). Une mesure est enregistrée toutes les 5 secondes. Expliquer le calcul.

Tournez la page S.V.P.

La tension en sortie de l'onduleur des panneaux solaires devrait être de 230 V \pm 10 %. Des valeurs qui sortent trop souvent de cet encadrement sont le signe d'un problème sur l'installation.

Question D8 | Créer une requête qui comptabilise le nombre de pics de sous-tension ou de surtension sur les 7 derniers jours avec `nom_appareil = « Onduleur panneaux solaires »` et `grandeur = « Tension AC »`

Question D9 | Créer une requête qui fournit une analyse détaillée des anomalies de tension de l'onduleur des panneaux solaires en fournissant un tableau contenant la date de l'anomalie, la valeur de la tension et le type d'anomalie (surtension ou sous-tension). Les résultats devront être classés par type d'anomalie (`nom_appareil = « Onduleur panneaux solaires »` et `grandeur = « Tension AC »`).

Paramétrage de Node-RED sur la Naiobox

La Naiobox est le composant central de la supervision du microgrid. Il permet de récupérer les informations envoyées par les différents constituants du microgrid et de stocker les données dans une base de données pour ensuite être consultées grâce à un logiciel SCADA implanté dans la Naiobox.

Comme les équipements communiquant avec la Naiobox utilisent des protocoles différents, le concepteur de la Naiobox a choisi de réaliser la collecte de ces données via l'utilisation du logiciel Node-RED : outil de programmation visuelle développé par IBM, principalement destiné à connecter des objets, services ou API entre eux via un système de flux de messages au format JSON (flows).

Objectif : Sécuriser le logiciel Node-RED sur environnement Linux, paramétrer la communication MQTT entre Node-RED et le M5Stack du local batterie.

La Naiobox contient un mini PC géré par un système d'exploitation Linux. Après installation du logiciel Node-RED, l'accès à l'interface de programmation de Node-RED, se fait par défaut en HTTP. Il est donc décidé de configurer le logiciel afin de n'autoriser que des connexions utilisant le protocole HTTPS.

Question E1 | Expliquez ce qu'est le protocole HTTPS et en quoi il est plus sécurisé que le protocole HTTP.

Dans un premier temps, afin de générer et stocker des certificats privés, on exécute sur la Naiobox les instructions suivantes :

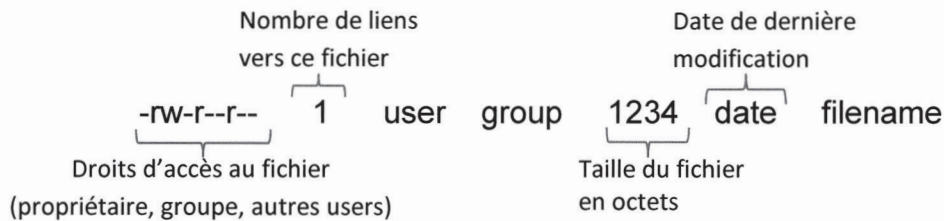
```
adminene@naiobox : ~ $ mkdir ssl
adminene@naiobox : ~ $ cd ssl
adminene@naiobox : ~/ssl $ openssl genrsa -out privatekey.pem 2048
adminene@naiobox : ~/ssl $ openssl req -new -sha256 -key privatekey.pem -out csr.pem
adminene@naiobox : ~/ssl $ openssl x509 -req -in csr.pem -signkey privatekey.pem -out certificate.pem
```

Question E2 | Indiquer ce que réalisent les deux premières instructions : mkdir ssl et cd ssl.

Les 3 dernières instructions permettent de générer un certificat (fichier **certificat.pem**) et une clé privée (**privatekey.pem**). Après l'exécution de celles-ci, on veut s'assurer qu'ils sont bien présents.

Question E3 | Donner l'instruction permettant de visualiser tous les fichiers (même cachés) avec leurs droits, présents dans le dossier **ssl** et ainsi vérifier que la clé privée a bien été générée.

On donne ci-dessous le format d'affichage des droits avec l'instruction demandée précédemment :



L'exécution de l'instruction précédente a permis d'obtenir le résultat ci-dessous.

```
adminene@naiobox:~/ssl$ pwd
/home/adminene/ssl
adminene@naiobox:~/ssl$ instruction à déterminer
total 20
drwxrwxr-x 2 adminene adminene 4096 oct.  3 13:02 .
drwxr-x--- 5 adminene adminene 4096 oct.  3 12:57 ..
-rw-rw-r-- 1 adminene adminene 1147 oct.  3 13:02 certificate.pem
-rw-rw-r-- 1 adminene adminene 1001 oct.  3 13:01 csr.pem
-rw----- 1 adminene adminene 1704 oct.  3 12:58 privatekey.pem
```

Figure E1 : Affichage des fichiers dans le dossier ssl

Question E4 | Résumer les droits en écriture et lecture affectés aux fichiers précédemment cités, les justifier par rapport à leur utilisation.

Pour activer et configurer la connexion en HTTPS dans Node-Red, il faut modifier un fichier nommé **settings.js**. Il s'agit d'un fichier javascript, comportant des propriétés d'objet JavaScript qu'il faut paramétrer.

On donne en DRE1 l'extrait du fichier **settings.js** relatif à la sécurité dans Node-RED. Dans cet extrait, une section permet de définir où Node-RED doit aller chercher la clé publique et le certificat, grâce à l'instruction `require("fs").readFileSync(chemin_fichier)`.

Question E5 | À partir de l'image figure E1, compléter la partie HTTPS du fichier **settings.js** dans DRE1, afin de donner le chemin de la clé privée et du certificat.

Par défaut, l'accès à Node-RED ne nécessite aucune identification. Il faut donc paramétrer un utilisateur administrateur (admin) dans le fichier **settings.js**. On génère alors le hash du mot de passe que l'on veut associer au compte admin via l'instruction : `Node-red admin hash-pw`.

On copie ensuite ce mot de passe dans la section `adminAuth` du fichier **settings.js** dans la propriété `password` (cf DRE1).

Question E6 | Donner la propriété principale d'un hash qui justifie son utilisation ici.

Le système à microcontrôleur M5Stack permet la mesure des conditions environnementales du local batterie. Il transmet ces informations à la Naiobox en utilisant le protocole MQTT.

Ce protocole est présenté brièvement en DTE1. Le broker (serveur) MQTT est installé sur la Naiobox.

Par exemple, toutes les 10 secondes, le M5Stack transmet à la Naiobox le taux de composants organiques volatiles (TVOC en anglais), qui peut permettre de détecter un début de défaillance des batteries. Cette communication est gérée en C++ dans le M5Stack notamment via l'instruction : `client.publish("localBatt/TVOC", payload, true)`. La variable `payload` contient l'information du taux de TVOC et l'horodatage de la mesure.

On souhaite paramétrer un nœud Node-RED permettant de collecter cette information.

Question E7 | Proposer un niveau de QOS pour cette communication en justifiant votre choix, puis compléter en DRE2 les champs sujet et QoS.

Une capture de trames a été effectuée au niveau de la Naiobox avec le logiciel Wireshark, puis on a appliqué un filtre de manière à ne conserver que les trames concernant le protocole MQTT. Une capture d'écran du logiciel Wireshark est donnée en DTE2.

Question E8 | En examinant les trames MQTT, donner la liste des grandeurs physiques mesurées dans le local batterie et indiquer à quelle fréquence sont effectuées ces mesures .

Question E9 | Identifier les adresses IP de la carte M5Stack et de la Naiobox. Compléter alors le champ serveur du bloc de configuration du nœud MQTT de Node-Red en DRE2 avec l'adresse complète du broker (IP + port).

Les règles de sécurité du broker MQTT sont définies dans son fichier de configuration. On peut notamment y paramétrer le port utilisé, l'autorisation de connexions non authentifiées, l'emplacement des éventuels mots de passe, les clés et certificats... On donne en figure E2 la copie du fichier de configuration utilisé lors des captures de trames.

```
# Place your local configuration in /etc/mosquitto/conf.d/  
# A full description of the configuration file is at  
# /usr/share/doc/mosquitto/examples/mosquitto.conf.example  
  
pid_file /run/mosquitto/mosquitto.pid  
allow_anonymous true  
password_file /etc/mosquitto/passwd  
listener 1883  
persistence true  
persistence_location /var/lib/mosquitto/  
  
log_dest file /var/log/mosquitto/mosquitto.log  
  
include_dir /etc/mosquitto/conf.d
```

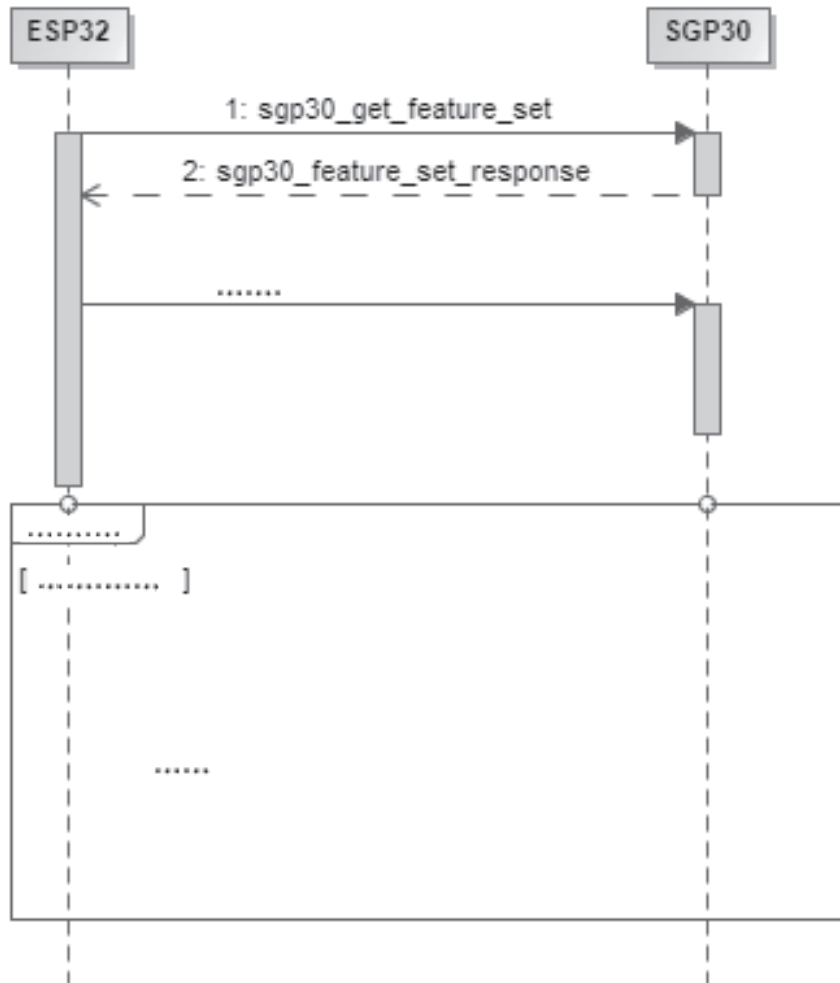
Figure E2 : Fichier de configuration du broker MQTT

Question E10 | Au vu de la figure E2 et du détail de trame MQTT, indiquer les failles dans la sécurité de la communication par MQTT.

NE RIEN ECRIRE DANS CE CADRE

DOCUMENT RÉPONSE DRB1 (Question B2)

Diagramme de séquence mesure COVT avec SGP 30





DOCUMENT RÉPONSE DRB2 (Questions B7 à B9)

Code de mesure du taux de COVT

Ligne	Code C++
1	<code>// Variables</code>
2	<code>uint16_t lastTvoc = 0; // dernière mesure de taux de COVT</code>
3	<code>unsigned long lastMeas = millis();</code>
4	
5	<code>While (true) { //boucle de mesure</code>
6	<code> unsigned long now = millis();</code>
7	<code> if (now - lastMeas >= 10000) { //mesure toutes les 10 s</code>
8	<code> lastMeas = now;</code>
9	
10	<code> if (sgp.IAQmeasure()) { // lancement de la mesure</code>
11	<code> // stockage du taux actuel mesuré, dans la variable tauxTvoc</code>
12	<code> </code>
13	
14	<code> // calcul du taux croissance par minute du taux de COVT</code>
15	<code> </code>
16	<code> </code>
17	<code> </code>
18	<code> //déclenchement de la ventilation et de l'alerte</code>
19	
20	<code> </code>
21	<code> </code>
22	<code> </code>
23	<code> </code>
24	<code> </code>
25	<code> </code>
26	
27	<code> else</code>
28	<code> {M5.Lcd.println("Erreur mesure !");}</code>
29	<code> }</code>
30	

NE RIEN ECRIRE DANS CE CADRE

DOCUMENT RÉPONSE DRC1 (Question C1 et C2)

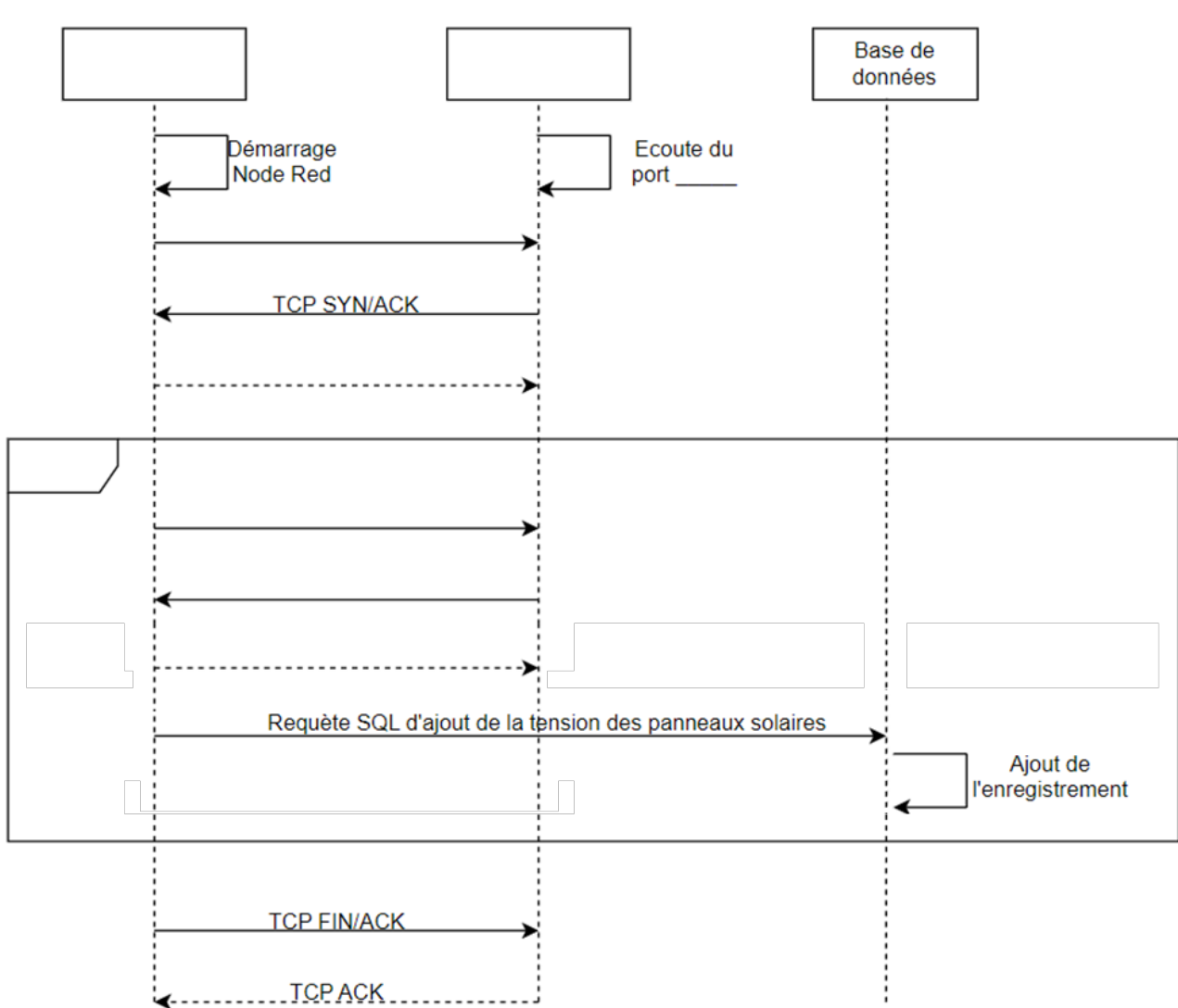
		
Nom	Onduleur panneaux solaires	Naiobox
Rôle (client ou serveur) :		
Adresse IPv4 :		
Masque de sous-réseau :		
Passerelle :		
Port :		

DOCUMENT RÉPONSE DRC2 (Question C4)

Modèle OSI	Modèle TCP/IP	Protocoles utilisés par le microgrid
7-		
6-		
5-		
4-		
3-		
2-		
1-		

NE RIEN ECRIRE DANS CE CADRE

DOCUMENT RÉPONSE DRC3 (Question C11)



Fichier de configuration Node-RED settings.js

```
/* *****  
 * Security  
 * - adminAuth  
 * - https  
 * - httpsRefreshInterval  
 * - requireHttps  
 * - httpNodeAuth  
 * - httpStaticAuth  
 * ***** /
```

```
/** To password protect the Node-RED editor and admin API, the following  
 * property can be used. See https://nodered.org/docs/security.html for details.*/
```

```
adminAuth: {  
  type: "credentials",  
  users: [{  
    username: "admin",  
password:"$2a$08$zWtXTja0fB1pzD4sHCMYOCMyz2Z6dNbM6tl8sJogENOMcxWV9DN.",  
    permissions: "*" } ]  
},
```

```
/** The following property can be used to enable HTTPS  
 * This property can be either an object, containing both a (private) key  
 * and a (public) certificate, or a function that returns such an object.  
 * See http://nodejs.org/api/https.html#https\_https\_createserver\_options\_requestlistener  
 * for details of its contents. */
```

```
/** Option 1: static object */
```

```
https: {  
  key: require("fs").readFileSync(.....), // à compléter  
  cert: require("fs").readFileSync(.....) // à compléter  
},
```

```
/** If the `https` setting is a function, the following setting can be used  
 * to set how often, in hours, the function will be called. That can be used  
 * to refresh any certificates. */
```

```
httpsRefreshInterval : 12,
```

```
/** The following property can be used to cause insecure HTTP connections to  
 * be redirected to HTTPS. */  
requireHttps: true,
```


NE RIEN ECRIRE DANS CE CADRE

Configuration nœud MQTT sur Node-RED

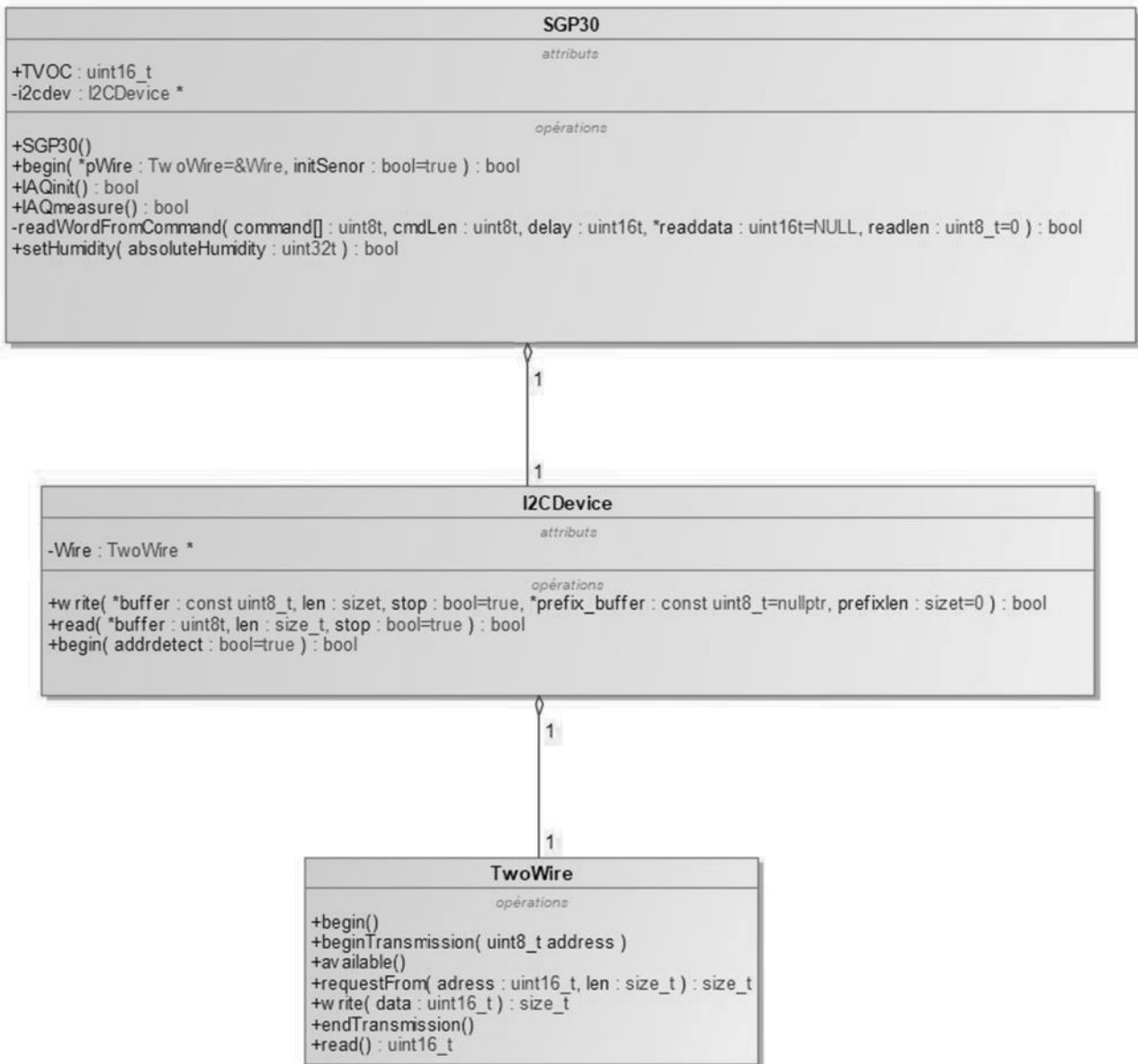
Supprimer Annuler Terminer

Propriétés ⚙️ 📄 🖨️

📍 Serveur	✎	+
Action	S'abonner à un seul sujet	▼	
📄 Sujet		
📶 QoS	▼	
👉 Sortie	détection automatique (objet JSON analysé, chaîne ou	▼	
🏷️ Nom	Nom		

DOCUMENT TECHNIQUE DTB1 (questions B5 à B7)

Structure de la classe SGP30



No.	Time	Source	Destination	Protocol	Length	Info
85	21.643484	192.168.8.1	192.168.8.4	TCP	74	41646 → 502 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2037001570 TSecr=0 WS=128
86	21.643895	192.168.8.4	192.168.8.1	TCP	74	502 → 41646 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM TSval=154180132 TSecr=2037001570
87	21.644498	192.168.8.1	192.168.8.4	TCP	66	41646 → 502 [ACK] Seq=1 Ack=1 Min=64256 Len=0 TSval=2037001571 TSecr=154180132
107	26.650606	192.168.8.1	192.168.8.4	Modbus/TCP	78	Query: Trans: 1; Unit: 1, Func: 3: Read Holding Registers
108	26.655130	192.168.8.4	192.168.8.1	Modbus/TCP	79	Response: Trans: 1; Unit: 1, Func: 3: Read Holding Registers
109	26.655631	192.168.8.1	192.168.8.4	TCP	66	41646 → 502 [ACK] Seq=13 Ack=14 Min=64256 Len=0 TSval=2037006582 TSecr=154185143
152	31.654942	192.168.8.1	192.168.8.4	Modbus/TCP	78	Query: Trans: 2; Unit: 1, Func: 3: Read Holding Registers
153	31.655846	192.168.8.4	192.168.8.1	Modbus/TCP	79	Response: Trans: 2; Unit: 1, Func: 3: Read Holding Registers
154	31.656095	192.168.8.1	192.168.8.4	TCP	66	41646 → 502 [ACK] Seq=25 Ack=27 Win=64256 Len=0 TSval=2037011583 TSecr=154190143
166	33.121067	192.168.8.1	192.168.8.4	TCP	66	41646 → 502 [FIN, ACK] Seq=25 Ack=27 Win=64256 Len=0 TSval=2037013048 TSecr=154190143
167	33.121486	192.168.8.4	192.168.8.1	TCP	66	502 → 41646 [ACK] Seq=27 Ack=26 Win=65280 Len=0 TSval=154191610 TSecr=2037013048

<p>> Frame 152: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface \Device\NPF_{CDC89913-7404-40B-Ethernet II, Src: RaspberryPII_20:ed:de (dc:a6:32:20:ed:de), Dst: MicroStarINT_e7:dd:dc (00:d8:61:e7:dd:dc)</p> <p>> Ethernet II, Src: RaspberryPII_20:ed:de (dc:a6:32:20:ed:de), Dst: MicroStarINT_e7:dd:dc (00:d8:61:e7:dd:dc)</p> <p>> Internet Protocol Version 4, Src: 192.168.8.1, Dst: 192.168.8.4</p> <p>> Transmission Control Protocol, Src Port: 41646, Dst Port: 502, Seq: 13, Ack: 14, Len: 12</p> <p>> Modbus/TCP</p> <p>Modbus</p> <p>.000 0011 = Function Code: Read Holding Registers (3)</p> <p>Reference Number: 40098</p> <p>Word Count: 2</p>	<pre> 0000 00 d8 61 e7 dd dc dc a6 32 20 ed de 08 00 45 00 ..a..... 2E. 0010 00 40 84 07 40 00 40 06 25 5b c0 a8 08 01 c0 a8 .@.@.%. 0020 08 04 a2 ae 01 f6 ca 29 85 a8 45 8b e4 9c 80 18 ) ..E..... 0030 01 f6 a2 3e 00 00 01 01 08 0a 79 6a 54 7e 09 30 .>.....]T~.0 0040 ad b7 00 02 00 00 06 01 03 9c a2 00 02 f..... </pre>
---	---

No.	Time	Source	Destination	Protocol	Length	Info
85	21.643484	192.168.8.1	192.168.8.4	TCP	74	41646 → 502 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2037001570 TSecr=0 WS=128
86	21.643895	192.168.8.4	192.168.8.1	TCP	74	502 → 41646 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM TSval=154180132 TSecr=2037001570
87	21.644498	192.168.8.1	192.168.8.4	TCP	66	41646 → 502 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2037001571 TSecr=154180132
✓	107 26.650606	192.168.8.1	192.168.8.4	Modbus/TCP	78	Query: Trans: 1; Unit: 1, Func: 3: Read Holding Registers
108	26.655130	192.168.8.4	192.168.8.1	Modbus/TCP	79	Response: Trans: 1; Unit: 1, Func: 3: Read Holding Registers
109	26.655631	192.168.8.1	192.168.8.4	TCP	66	41646 → 502 [ACK] Seq=13 Ack=14 Win=64256 Len=0 TSval=2037006582 TSecr=154185143
152	31.654942	192.168.8.1	192.168.8.4	Modbus/TCP	78	Query: Trans: 2; Unit: 1, Func: 3: Read Holding Registers
153	31.655846	192.168.8.4	192.168.8.1	Modbus/TCP	79	Response: Trans: 2; Unit: 1, Func: 3: Read Holding Registers
154	31.656095	192.168.8.1	192.168.8.4	TCP	66	41646 → 502 [ACK] Seq=25 Ack=27 Win=64256 Len=0 TSval=2037011583 TSecr=154190143
166	33.121067	192.168.8.1	192.168.8.4	TCP	66	41646 → 502 [FIN, ACK] Seq=25 Ack=27 Win=64256 Len=0 TSval=2037013048 TSecr=154190143
167	33.121486	192.168.8.4	192.168.8.1	TCP	66	502 → 41646 [ACK] Seq=27 Ack=26 Win=65280 Len=0 TSval=154191610 TSecr=2037013048

Détail trame n°108

```

Sequence Number (raw): 1166795919
[Next Sequence Number: 14 (relative sequence number)]
Acknowledgment Number: 13 (relative ack number)
Acknowledgment number (raw): 3391718824
1000 .... = Header Length: 32 bytes (8)
v Flags: 0x018 (PSH, ACK)
000. .... = Reserved: Not set
.... .... = Accurate ECN: Not set
.... 0... = Congestion Window Reduced: Not set
.... 0... = ECN-Echo: Not set
.... 0... = Urgent: Not set
.... 1... = Acknowledgment: Set
.... 1... = Push: Set
.... 0... = Reset: Not set
.... 0... = Syn: Not set
.... 0... = Fin: Not set

```

```

0000 dc a6 32 20 ed de 00 d8 61 e7 dd dc 08 00 45 00 ..2 .... a .....E
0010 00 41 76 30 40 00 80 06 f3 30 c0 a8 08 04 c0 a8 -Av0@... 0 .....
0020 08 01 01 f6 a2 ae 45 8b e4 8f ca 29 85 a8 80 18 .....E ...
0030 00 ff 50 5f 00 00 01 01 08 0a 09 30 ad b7 79 6a ..P.....0..yJ
0040 40 f1 00 01 00 00 00 07 01 03 04 14 00 ff fe @.....

```

No.	Time	Source	Destination	Protocol	Length	Info
85	21.643484	192.168.8.1	192.168.8.4	TCP	74	41646 → 502 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2037001570 TSecr=0 WS=128
86	21.643895	192.168.8.4	192.168.8.1	TCP	74	502 → 41646 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM TSval=154180132 TSecr=2037001570
87	21.644498	192.168.8.1	192.168.8.4	TCP	66	41646 → 502 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2037001571 TSecr=154180132
107	26.650606	192.168.8.1	192.168.8.4	Modbus/TCP	78	Query: Trans: 1; Unit: 1, Func: 3: Read Holding Registers
108	26.655130	192.168.8.4	192.168.8.1	Modbus/TCP	79	Response: Trans: 1; Unit: 1, Func: 3: Read Holding Registers
109	26.655631	192.168.8.1	192.168.8.4	TCP	66	41646 → 502 [ACK] Seq=13 Ack=14 Win=64256 Len=0 TSval=2037006582 TSecr=154185143
152	31.654942	192.168.8.1	192.168.8.4	Modbus/TCP	78	Query: Trans: 2; Unit: 1, Func: 3: Read Holding Registers
153	31.655846	192.168.8.4	192.168.8.1	Modbus/TCP	79	Response: Trans: 2; Unit: 1, Func: 3: Read Holding Registers
154	31.656095	192.168.8.1	192.168.8.4	TCP	66	41646 → 502 [ACK] Seq=25 Ack=27 Win=64256 Len=0 TSval=2037011583 TSecr=154190143
166	33.121067	192.168.8.1	192.168.8.4	TCP	66	41646 → 502 [FIN, ACK] Seq=25 Ack=27 Win=64256 Len=0 TSval=2037013048 TSecr=154190143
167	33.121486	192.168.8.4	192.168.8.1	TCP	66	502 → 41646 [ACK] Seq=27 Ack=26 Win=65280 Len=0 TSval=1541916110 TSecr=2037013048

[Bytes in flight: 13]
 [Bytes sent since last PSH flag: 13]
 TCP payload (13 bytes)
 [POU Size: 13]
 Modbus/TCP

Transaction Identifier: 1
 Protocol Identifier: 0
 Length: 7
 Unit Identifier: 1
 Modbus

.000 0011 = Function Code: Read Holding Registers (3)
[\[Request Frame: 107\]](#)
 Byte Count: 4
 > Register 40098 (INT16): 5120
 > Register 40099 (INT16): -2


```

0000  dc a6 32 20 ed de 00 d8 61 e7 dd dc 08 00 45 00  ..2....a.....E
0010  00 41 76 30 40 00 80 06 f3 30 c0 a8 08 04 c0 a8  ..Av@... 0.....
0020  08 01 01 f6 a2 ae 45 8b e4 8f ca 29 85 a8 80 18  .....E....)....
0030  00 ff 50 5f 00 00 01 01 08 0a 09 30 ad b7 79 6a  ..P_.....0..y..
0040  40 f1 00 01 00 00 00 07 01 03 04 14 00 ff fe  @.....
    
```

DOCUMENT TECHNIQUE DTC4

Inverter Model MODBUS Register Mappings

The following table lists the supported MODBUS register values. Unsupported values are indicated by the NOT_IMPLEMENTED value. The base register of the Device Specific block is set to 40070 (MODBUS PLC address [base 1]), or 40069 (MODBUS Protocol Address [base 0]).

- **acc32** is a uint32 accumulator that should always increase. Its value is in the range of 0...4294967295.
- **Scale Factors.** As an alternative to floating point format, values are represented by Integer values with a signed scale factor applied. The scale factor explicitly shifts the decimal point to left (negative value) or to the right (positive value).

For example, a value “Value” may have an associated value “Value_SF”

Value = “Value” * 10^{Value_SF} for example:

- For “Value” = 2071 and “Value_SF” = -2 Value = 2071*10⁻² = 20.71
- For “Value” = 2071 and “Value_SF” = 2 Value = 2071*10² = 20710

Address		Size	Name	Type	Units	Description
(base 0)	(base 1)					
40091	40092	1	I_AC_PF ¹	int16	%	Power Factor
40092	40093	1	I_AC_PF_SF ¹	int16		Scale factor
40093	40094	2	I_AC_Energy_WH	acc32	WattHours	AC Lifetime Energy production
40095	40096	1	I_AC_Energy_WH_SF	uint16		Scale factor
40096	40097	1	I_DC_Current	uint16	Amps	DC Current value
40097	40098	1	I_DC_Current_SF	int16		Scale factor
40098	40099	1	I_DC_Voltage	uint16	Volts	DC Voltage value
40099	40100	1	I_DC_Voltage_SF	int16		Scale factor
40100	40101	1	I_DC_Power	int16	Watts	DC Power value
40101	40102	1	I_DC_Power_SF	int16		Scale factor
40103	40104	1	I_Temp_Sink	int16	Degrees C	Heat Sink Temperature
40106	40107	1	I_Temp_SF	int16		Scale factor
40107	40108	1	I_Status	uint16		Operating State
40108	40109	1	I_Status_Vendor	uint16		Vendor-defined operating state and error codes. For error description, meaning and troubleshooting, refer to the SolarEdge Installation Guide.

Fiche résumée des fonctions et mots clés SQL usuels

Description	Syntaxe SQL
Créer une base de données	CREATE DATABASE nom de la base;
Créer une table dans une base de données	CREATE TABLE nomTable (id INT NOT NULL AUTO_INCREMENT , champ1 DOUBLE , champ2 VARCHAR(255) , champ3 TIMESTAMP DEFAULT NOW() , PRIMARY KEY(id));
Écrire une nouvelle entrée dans une table de BDD	INSERT INTO nomTable(champ1, champ2, champ3) VALUES ('valeur1', 'valeur2', NOW());
Modifier les informations de l'entrée dont le champ id = 51	UPDATE nomTable SET nomChamp1=10, nomChamp2=32 WHERE id=51;
Sélectionner toutes les informations de la table	SELECT * FROM nomTable;
Sélectionner seulement les informations d'un champ	SELECT nomChamp FROM nomTable;
Sélectionner tous les champs de la table correspondant à deux critères	SELECT * FROM nomTable WHERE nomChamp1 = 'poste' AND nomChamp3 < 12;
Sélectionner sur plusieurs tables (jointure)	SELECT * FROM nomTable1 JOIN nomTable2 ON nomTable1.nomChamp1 = nomTable2.nomChamp4;
Appliquer une fonction d'agrégation (MIN , MAX , COUNT , AVG) sur des valeurs d'un champ regroupées	SELECT MAX (nomChamp1) FROM nomTable GROUP BY nomChamp2;
Utiliser DATE() pour extraire la partie date	SELECT DATE (nomChamp3) FROM nomTable;
Utiliser NOW() pour insérer la date et l'heure actuelles	INSERT INTO nomTable(champ1, champ2, champ3) VALUES ('valeur1', 'valeur2', NOW());
Utiliser CASE pour des conditions personnalisées	SELECT nomChamp1, CASE WHEN nomChamp1 > 100 THEN 'Élevé' ELSE 'Faible' END AS categorie FROM nomTable;
Ordonner les résultats d'une requête par ordre croissant ou décroissant	SELECT * FROM nomTable ORDER BY nomChamp1 ASC ;
Filtrer par intervalle de dates	SELECT * FROM nomTable WHERE champ3 BETWEEN '2025-01-01' AND NOW() ;

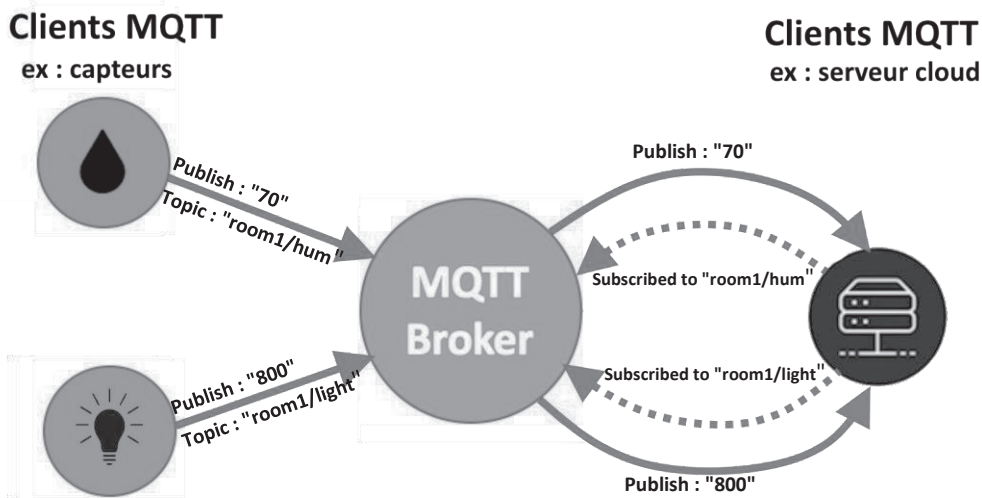
Les mots en gras dans la colonne de droite sont des mots réservés par le langage SQL.

Protocole MQTT

MQTT (Message Queuing Telemetry Transport) est un protocole de messagerie léger s'appuyant sur la suite de protocoles TCP/IP. Il est particulièrement utile pour les applications IoT (Internet des Objets). Il est conçu pour être fiable, simple, léger en bande passante.

Fonctionnement

Un client va publier des données sur un sujet (topic) et envoyer ces données au broker (serveur). Le broker, va renvoyer ces données aux clients abonnés (subscribers) ayant préalablement fait la demande d'abonnement pour ce sujet (topic) au broker .



Un topic est une chaîne de caractère. Les topics peuvent avoir une hiérarchie qui permet de sélectionner finement les informations que l'on désire en séparant les niveaux de sujets par un « / ».

Exemple de structuration des topics : contexte/emplacement/capteur ou action/type...

evlink/borne1/energie

evlink/borne1/cmd

evlink/borne1/# (le # signifie tous les topics du sous niveau considéré)

Qualité de Service

Le protocole MQTT offre trois niveaux de Qualité de Service (QoS) pour garantir la fiabilité de la livraison des messages.

- QoS 0 (Au plus une fois) : Le message est envoyé une seule fois sans confirmation. Pas de garantie.
- QoS 1 (Au moins une fois) : Le message est ré-envoyé jusqu'à accusé de réception. Risque de doublons.
- QoS 2 (Exactement une fois) : Le message est livré une seule fois et garanti sans doublon grâce à un échange en quatre étapes. Plus coûteux en temps.

Le paramétrage de ce QoS s'effectue à l'envoi d'un message lors d'une commande publish et lors de l'abonnement (subscribe). Le QoS effectif sera le minimum entre le QoS demandé par le client à l'abonnement et celui utilisé lors de la publication.

Tournez la page S.V.P.

DTE2 (Question E8 à E10)

Listing des trames MQTT

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.8.12	192.168.8.1	MQTT	93	Publish Message [localBatt/TVOC]
3	0.347017	192.168.8.12	192.168.8.1	MQTT	95	Publish Message [localBatt/eCO2]
5	6.208854	192.168.8.12	192.168.8.1	MQTT	56	Ping Request
7	6.209112	192.168.8.1	192.168.8.12	MQTT	56	Ping Response
9	10.078543	192.168.8.12	192.168.8.1	MQTT	93	Publish Message [localBatt/TVOC]
11	10.345032	192.168.8.12	192.168.8.1	MQTT	95	Publish Message [localBatt/eCO2]
13	18.922619	192.168.8.12	192.168.8.1	MQTT	101	Publish Message [localBatt/humidite]
15	19.323261	192.168.8.12	192.168.8.1	MQTT	104	Publish Message [localBatt/temperature]
17	19.846324	192.168.8.12	192.168.8.1	MQTT	93	Publish Message [localBatt/TVOC]
19	20.353539	192.168.8.12	192.168.8.1	MQTT	95	Publish Message [localBatt/eCO2]
21	21.218737	192.168.8.12	192.168.8.1	MQTT	56	Ping Request
23	21.219185	192.168.8.1	192.168.8.12	MQTT	56	Ping Response
25	29.850236	192.168.8.12	192.168.8.1	MQTT	93	Publish Message [localBatt/TVOC]
27	30.353109	192.168.8.12	192.168.8.1	MQTT	95	Publish Message [localBatt/eCO2]
29	36.226733	192.168.8.12	192.168.8.1	MQTT	56	Ping Request
31	36.226984	192.168.8.1	192.168.8.12	MQTT	56	Ping Response
33	39.848689	192.168.8.12	192.168.8.1	MQTT	93	Publish Message [localBatt/TVOC]
35	40.351027	192.168.8.12	192.168.8.1	MQTT	95	Publish Message [localBatt/eCO2]
37	48.904532	192.168.8.12	192.168.8.1	MQTT	101	Publish Message [localBatt/humidite]
39	49.324926	192.168.8.12	192.168.8.1	MQTT	104	Publish Message [localBatt/temperature]
41	49.851230	192.168.8.12	192.168.8.1	MQTT	93	Publish Message [localBatt/TVOC]
43	50.350288	192.168.8.12	192.168.8.1	MQTT	95	Publish Message [localBatt/eCO2]
45	51.242186	192.168.8.12	192.168.8.1	MQTT	56	Ping Request
47	51.242321	192.168.8.1	192.168.8.12	MQTT	56	Ping Response
49	59.961886	192.168.8.12	192.168.8.1	MQTT	94	Publish Message [localBatt/TVOC]
51	60.369761	192.168.8.12	192.168.8.1	MQTT	95	Publish Message [localBatt/eCO2]
53	66.312573	192.168.8.12	192.168.8.1	MQTT	56	Ping Request
55	66.312823	192.168.8.1	192.168.8.12	MQTT	56	Ping Response

Détail de la trame n° 39

```

> Frame 39: 104 bytes on wire (832 bits), 104 bytes captured (832 bits)
> Ethernet II, Src: Espressif_69:d6:a8 (a0:dd:6c:69:d6:a8), Dst: RaspberryPiT_93:95:da (d8:3a:dd:93:95:da)
> Internet Protocol Version 4, Src: 192.168.8.12, Dst: 192.168.8.1
> Transmission Control Protocol, Src Port: 62629, Dst Port: 1883, Seq: 551, Ack: 7, Len: 50

```

```

MQ Telemetry Transport Protocol, Publish Message
  > [Expert Info (Note/Protocol): Unknown version (missing the CONNECT packet?)]
  > Header Flags: 0x30, Message Type: Publish Message, QoS Level: At most once delivery (Fire and Forget)
    Msg Len: 48
    Topic Length: 21
    Topic: localBatt/temperature
    Message: 32342e36307c323032352d31302d30342031363a31353a3433

```

```

0000  d8 3a dd 93 95 da a0 dd 6c 69 d6 a8 08 00 45 00  ..:..... li....E.
0010  00 5a 00 3d 00 00 ff 06 2a 03 c0 a8 08 0c c0 a8  .Z.=.... *.....
0020  08 01 f4 a5 07 5b 05 f6 84 e2 7a 95 06 22 50 18  .....[... ..z.. "P.
0030  16 5a 87 ae 00 00 30 30 00 15 6c 6f 63 61 6c 42  .Z....00 ..localB
0040  61 74 74 2f 74 65 6d 70 65 72 61 74 75 72 65 32  att/temperature 2
0050  34 2e 36 30 7c 32 30 32 35 2d 31 30 2d 30 34 20  4.60|202 5-10-04
0060  31 36 3a 31 35 3a 34 33                          16:15:43

```