

Extrait de sujet

Partie A. Analyse du contexte

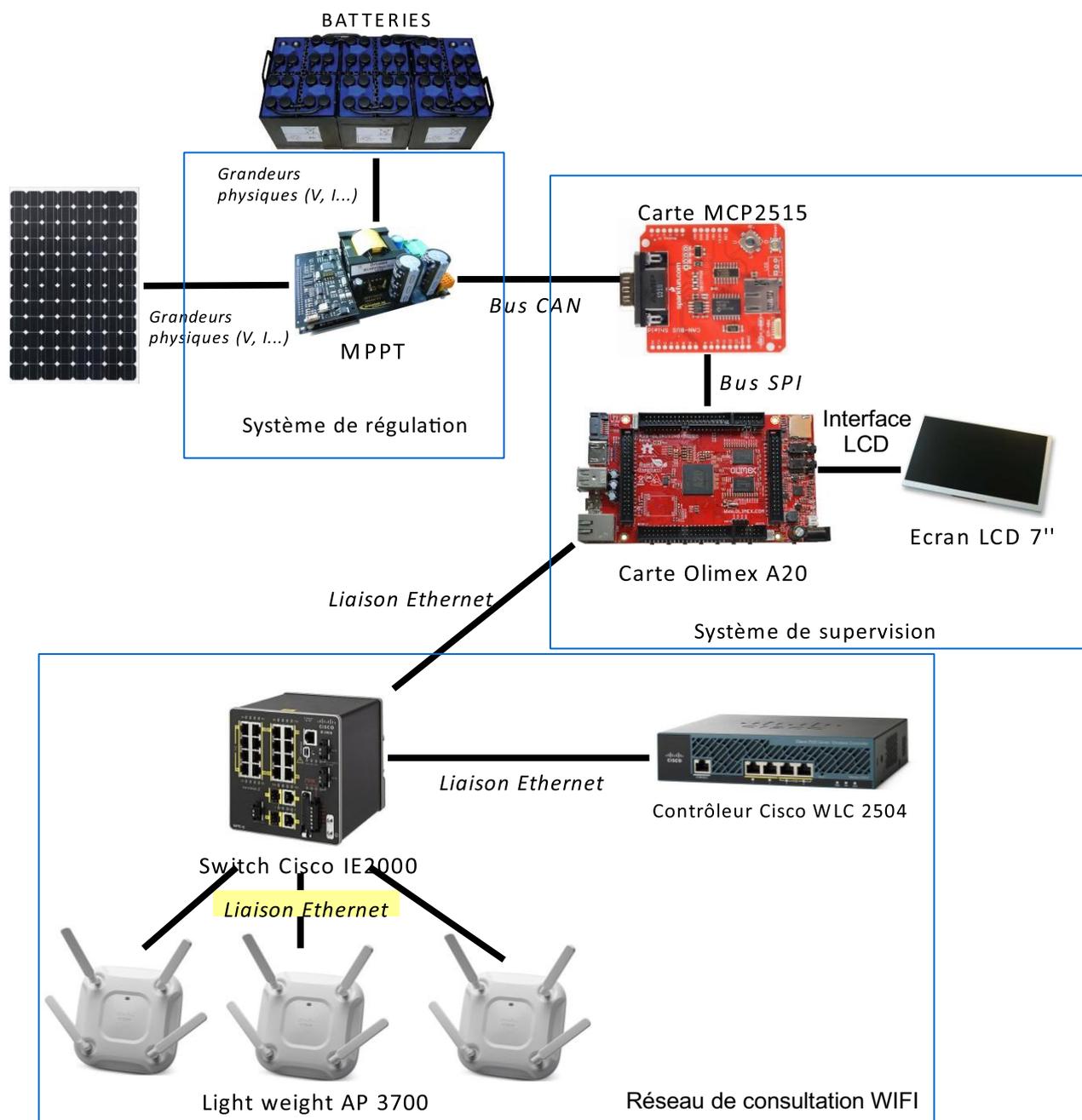


Figure 1 : synoptique du système étudié

Le *MPPT* permet de récupérer les informations sur l'état de charge des batteries et la production d'énergie électrique. Il communique ces informations au système de supervision par le bus CAN.

Le système de supervision est constitué de la carte MCP2515, qui est un contrôleur CAN, et de la carte Olimex.

La carte *Olimex* offre quatre services :

- récupération des informations en provenance du bus CAN via la carte MCP2515 ;
- stockage des informations dans une base de données locale ;
- affichage des mesures sur un écran LCD ;
- hébergement d'un site Web qui présente aux passagers les informations sur la production électrique des panneaux solaires.

Le réseau WIFI est constitué de neuf points d'accès *Light weight Acces Point 3700* (LAP) : trois par wagon.

Ces points d'accès sont contrôlés par un contrôleur de LAP (WLC 2504). Les passagers peuvent se connecter au réseau WIFI avec leur équipement personnel (smartphone / tablette / PC portable) et ainsi charger la page Web hébergée sur la carte « *Olimex* ».

Lorsqu'un passager est connecté en WIFI, il est lié au point d'accès le plus proche, en accord avec le contrôleur WLC2504.

LES CAS D'UTILISATION

Les cas présentés ci-dessous ne couvrent que la supervision des mesures énergétiques.

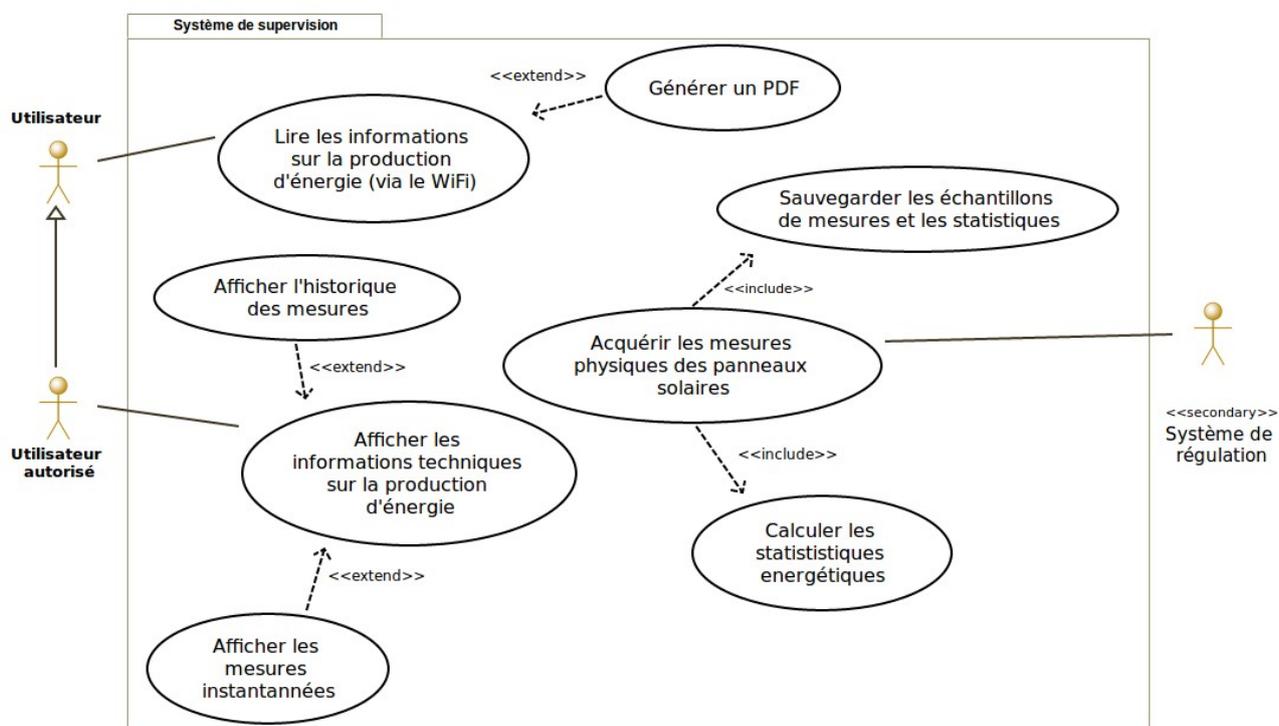


Figure 2 : diagramme de cas d'utilisation

Un « Utilisateur » est, par exemple, un passager du train, alors qu'un « Utilisateur autorisé », comme un technicien SNCF, possède plus de droits.

Il existe trois cas d'utilisation principaux :

- « Lire les informations sur la production d'énergie », qui permet de visualiser les informations sur la production d'énergie via une page Web adaptée au grand public ;
- « Afficher les informations techniques sur la production d'énergie », qui permet d'afficher sous une forme plus technique les informations contenues dans la base de données, à destination de l'utilisateur autorisé ;

- « Acquérir les mesures physiques des panneaux solaires », qui permet d'acquérir les informations et de les sauvegarder dans la base de données.
Le stockage des informations de mesure est réalisé dans une base de données qui est hébergée sur la carte « Olimex » ;

Avant l'enregistrement, il est nécessaire de calculer l'énergie produite par les panneaux solaires.

Partie B. Étude du système de supervision de la production d'énergie

L'objectif de cette partie est d'analyser et de compléter la spécification technique du système de supervision, notamment à travers la modélisation UML.

À l'aide du synoptique du système Figure 1 et du diagramme de cas d'utilisation Figure 2, répondre aux questions concernant la topologie du système de supervision

Q1. Pour chaque question, cocher la case « VRAI » ou « FAUX ».

Propositions	VRAI	FAUX
Cette représentation est la description du modèle vu par les acteurs du système	X	
Le technicien SNCF, qui est un utilisateur autorisé, a un contrôle sur la production d'énergie		X
D'un point de vue UML l'utilisateur est une spécialisation de l'utilisateur autorisé (technicien)		X
Le technicien SNCF, qui est un utilisateur autorisé, peut modifier les informations concernant la production d'énergie à destination des passagers		X

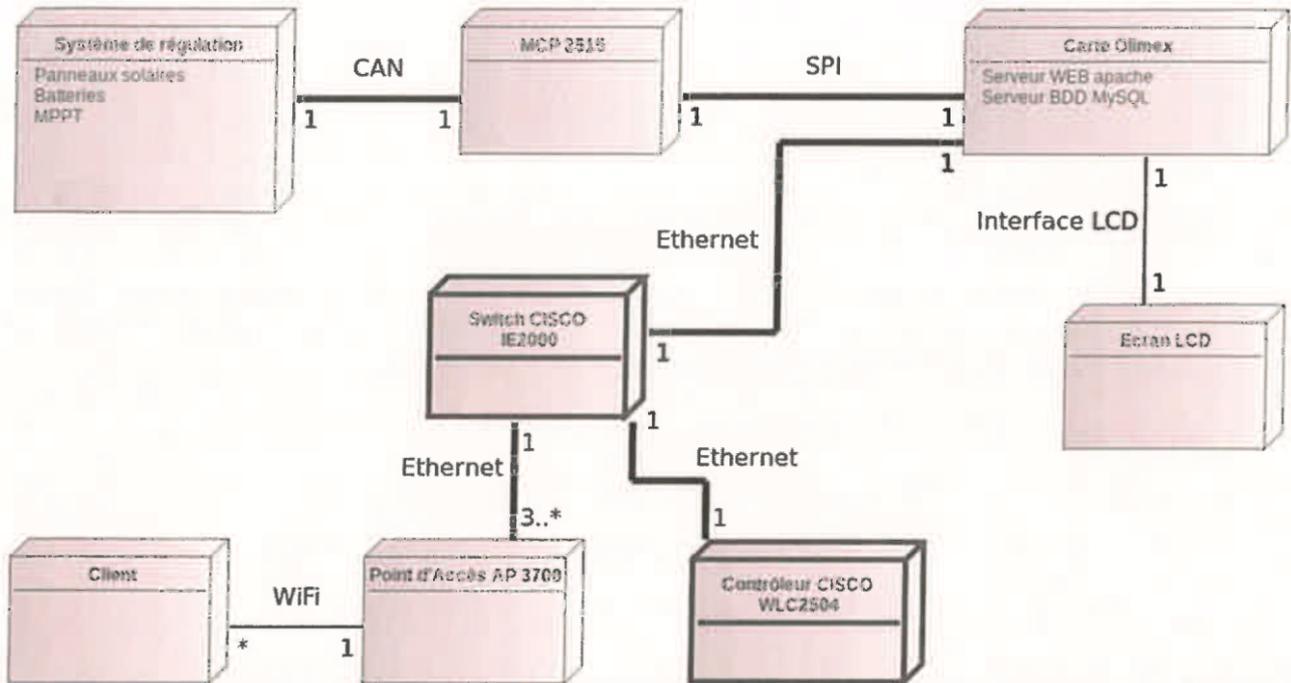
Q2. Justifier la présence de la relation « include » entre les cas d'utilisation « Sauvegarder les échantillons de mesures et les statistiques » et « Acquérir les mesures physiques des panneaux solaires ».

Il n'est pas possible de sauvegarder les informations sans en avoir fait l'acquisition. "Acquérir les mesures physiques des panneaux solaires" inclus bien le cas "sauvegarder les échantillons de mesures et les statistiques".

Q3. Justifier la relation « extend » entre les cas d'utilisation « Générer un PDF » et « Lire les informations sur la production d'énergie ».

La génération du PDF n'a rien d'obligatoire, c'est une option supplémentaire pour le cas "Lire les informations sur la production d'énergie via le WIFI".

Q4. Compléter les liaisons, les composants manquants, les cardinalités et les noms des supports transportant l'information dans le diagramme de déploiement.



Partie C. Étude de la communication entre le MPPT et le système de supervision

L'objectif de cette partie est de valider l'architecture de communication sur le bus CAN entre le MPPT et le système de supervision.

Le MPPT mesure les caractéristiques (tension et courant) de l'énergie produite par les panneaux solaires, la tension des batteries et la température du système.

Ces informations sont encapsulées dans une trame CAN à destination du système de supervision.

L'échange d'information entre le MPPT et le système de supervision fonctionne selon un schéma trame de requête / trame de données comme suit :

- le système de supervision envoie une **trame de requête** ;
- le MPPT répond à cette requête en renvoyant une **trame de données** contenant les informations sur la production d'énergie.

La trame de requête est émise toutes les 30 minutes.

*L'ensemble des informations nécessaires pour répondre à cette partie se trouve dans les **documentations PP1 et PP2**.*

Le format des trames CAN est présenté en **documentation PP2**.

Q5. Indiquer le rôle des champs SOF(Start Of Frame), EOF(End Of Frame) et CRC dans une trame CAN.

SOF: début de la trame

EOF: fin de la trame

CRC: champ correcteur d'erreur, calculé à partir de tous les champs émis avant celui-ci.

Afin d'acquérir les données du MPPT, le système de supervision doit émettre une **trame de requête CAN** avec « MPPT ID » = 0001 et 7 octets de données attendus.

Q6. Compléter en binaire le contenu de la trame de requête CAN, tout en faisant apparaître les différents champs.

SOF	ID	RTR	RES	DLC	CRC (16 bits)	ACK	EOF
0	1110001 0001	1	00	0111		11	1111111

Le MPPT mesure la tension et le courant en provenance des panneaux solaires (U_{IN} et I_{IN}), la tension en provenance des batteries (U_{OUT}) et la température du système ($T_{AMBIANTE}$).

Q7. À partir de la documentation PP1, relever les tensions typiques de fonctionnement, en entrée et en sortie du MPPT, le courant maximum en entrée du MPPT, les températures maximum et minimum de fonctionnement du MPPT.

$U_{IN} = 24V$ (typique)

$I_{IN} = 150A$ (maximum)

$T_{ambiante} = -30^{\circ}C$ à $70^{\circ}C$ (minimum/maximum)

Les mesures effectuées par le MPPT sont numérisées puis encapsulées dans une **trame de données CAN**.

On a capturé une trame de données émise par le MPPT. Le contenu du champ DATA de cette trame est le suivant :

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 1	0	0	0	1	0	0	1	1
Byte 2	0	1	1	0	1	1	0	1
Byte 3	0	0	0	0	0	0	0	0
Byte 4	1	0	0	0	1	1	1	1
Byte 5	0	0	0	0	0	0	1	1
Byte 6	0	1	1	0	1	1	0	1
Byte 7	0	0	0	1	1	0	0	1

Q8. Préciser les valeurs de U_{in}, I_{in}, U_{out} et température ambiante sous forme hexadécimale et décimale, et calculer les grandeurs correspondantes. Utiliser la **documentation PP1** pour la mise à l'échelle des valeurs lues.

```

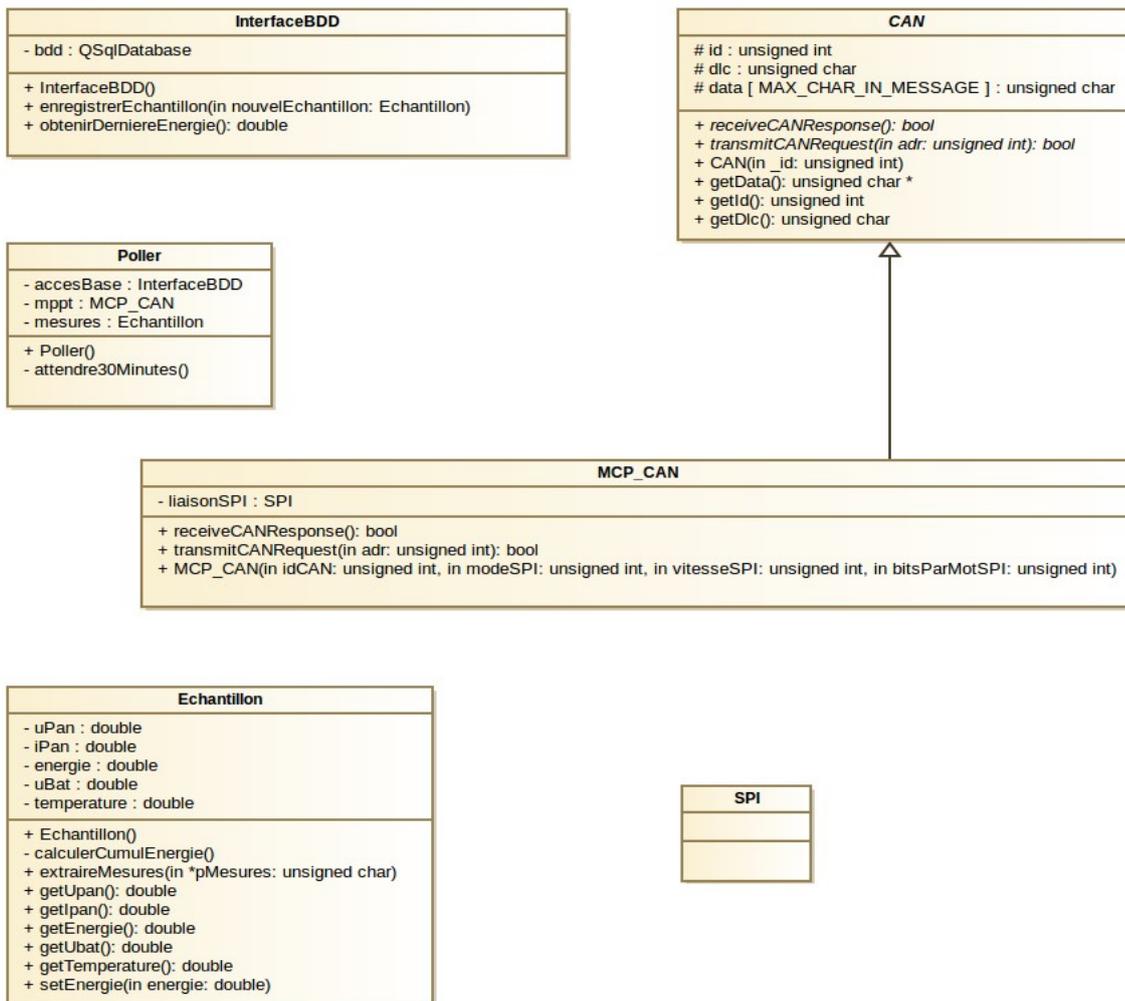
Tambiante = 0001 1001 = 0x19 = 25°C
UOUT = 11 0110 1101 = 0x36D = 877 = 28x877 / 1023 = 24V
IIN = 00 1000 1111 = 0x08F = 143 = 150x143 / 1023 = 20,97A
UIN = 11 0110 1101 = 0x36D = 877 = 28x877 / 1023 = 24V
    
```

Partie D. Étude des services d'acquisition et de stockage des informations

Cette partie concerne le processus d'acquisition des mesures de production par le système de supervision et leur enregistrement dans la base de données.

Elle a pour objectif de contribuer à sa modélisation, de valider son architecture logicielle, et de proposer des améliorations.

Le diagramme de classe permettant de remplir cette tâche est donné ci-dessous.



Au démarrage du processus, un objet de la classe Poller est instancié. La base de données est ouverte et une lecture de la dernière valeur de l'énergie est réalisée.

Ensuite, une requête est envoyée au MPPT toutes les trente minutes. La réponse à cette requête contient les informations sur un échantillon de mesure énergétique.

Ce dernier est mis en forme et enregistré dans la BDD.

La classe « Echantillon » permet de gérer un échantillon de mesures.

Elle contient les attributs suivants :

- *« uPan » et « iPan » qui sont respectivement la tension et l'intensité délivrées par le panneau solaire. Ces attributs correspondent aux valeurs U_{in} et I_{in} du MPPT ;*
- *« energie » qui correspond à l'énergie cumulée ;*
- *« uBat » pour la tension de la batterie. Cet attribut correspond à la valeur U_{out} du MPPT ;*
- *« temperature » qui correspond à la température ambiante.*

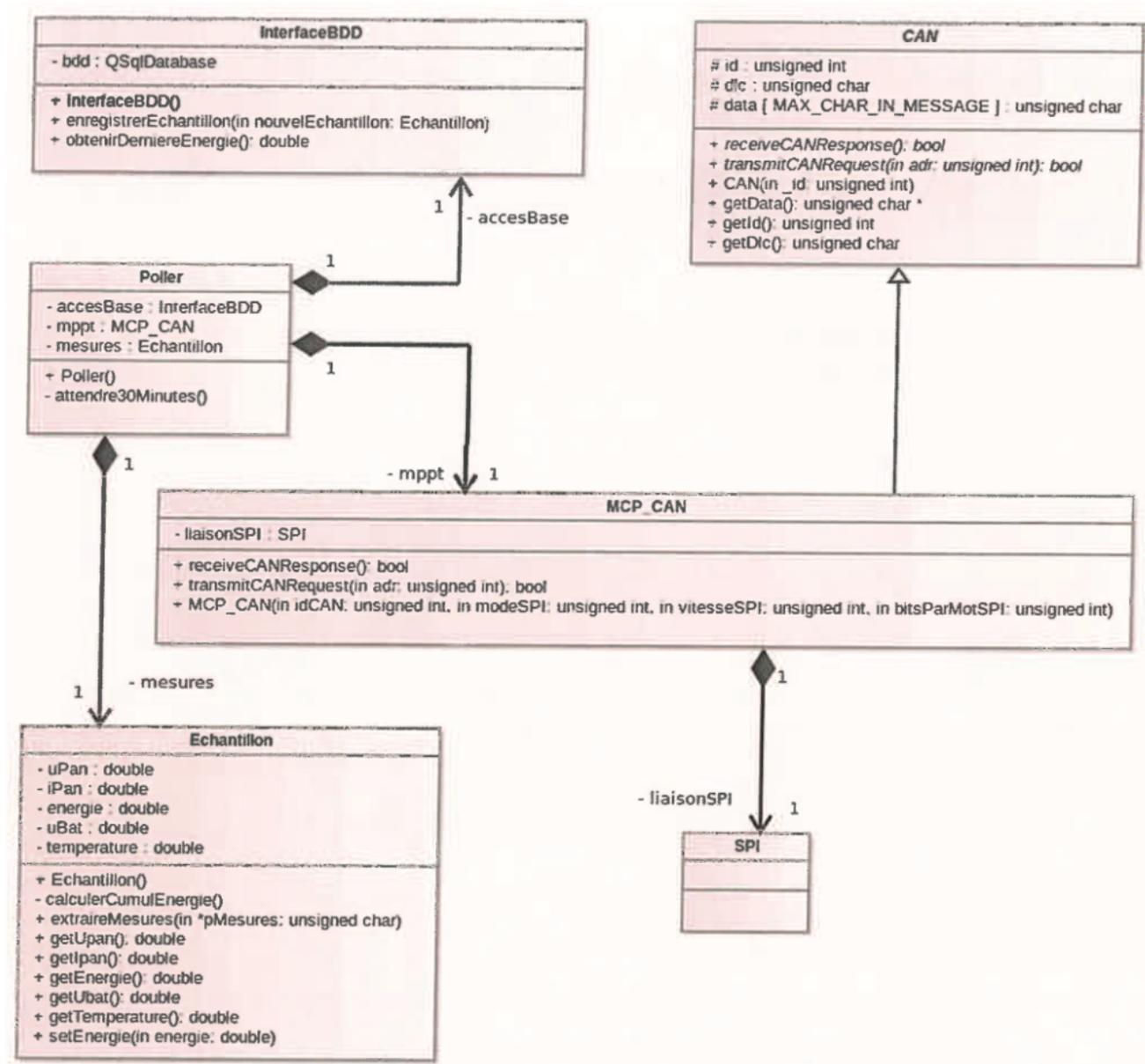
Q9. Indiquer et justifier la relation entre la classe CAN et la classe MCP_CAN.

Il s'agit d'un héritage. MCP_CAN est un CAN particulier, intégrant une communication SPI.

Q10. Indiquer si une instance de MCP_CAN peut (ou ne peut pas) accéder à l'attribut « data » déclaré dans la classe CAN. Justifier votre réponse.

L'attribut data est protégé (#: protected), il est donc visible et utilisable par la classe CAN et les classes dérivées (classes filles / sous-classes) MCP_CAN.

Q11. Compléter les associations, les noms de rôles et les cardinalités sur le diagramme de classes.



Les questions suivantes concernent la classe « Poller ».

Le constructeur de la classe « Poller » effectue une boucle de scrutation de période 30 minutes.

À chaque tour de boucle, on envoie une requête sur le bus CAN, on attend la réponse du MPPT, on traite les informations dans la classe « Echantillon » puis on stocke ces informations dans la base de données.

Q12. Indiquer la méthode et la classe à appeler pour envoyer une requête au MPPT.

transmitCANRequest de la classe MCP_CAN

Q13. Indiquer la méthode et la classe à appeler pour enregistrer un échantillon dans la base de données.

enregistrerEchantillon de la classe InterfaceBDD

Q14. Indiquer la méthode et la classe à appeler pour extraire les mesures issues du MPPT et les stocker dans l'échantillon.

extraireMesures de la classe Echantillon

Q15. Dans le document réponses, compléter en langage C++, le constructeur de la classe « Poller », en vous aidant du diagramme de séquence en **documentation PP3**.

```
Poller::Poller()
{
    double cumulEnergie =
    accesBase.obtenirDerniereEnergie();
    mesures.setEnergie(cumulEnergie) ;      while(true)
    {
    bool retReceive=false;
    bool retTransmit=false;
    unsigned char *pData=NULL;
```

```
    // boucles pour réception
    while(retReceive == false)
    {
        while (retTransmit == false)
        {
            retTransmit = mppt.transmitCANRequest(0x711) ;
        }
        retReceive = mppt.receiveCANResponse() ;
    }

    // récupération des données
    pData = mppt.getData() ;

    // extraction des mesures
    mesures.extraireMesure(pData) ;

    // enregistrement de l'échantillon dans la bdd
    accesBase.enregistrerEchantillon(mesures) ;

    // attente période de scrutation
    attendre30Minutes() ;
```

```
    }
}
```

Les questions suivantes concernent les classes MCP_CAN et Echantillon.

Q16. Écrire en C++ la déclaration de la classe MCP_CAN

```
class MCP_CAN : public CAN
{
public:

    MCP_CAN(unsigned int idCAN,unsigned int modeSPI,
            unsigned int vitesseSPI,unsigned int bitParMotSPI) ;

    ~MCP_CAN();
    bool receiveCANResponse() ;
    bool transmitCANRequest(unsigned int adr) ;
private:
    SPI liaisonSPI ;
} ;
```

La classe « Echantillon » permet d’instancier un objet contenant les valeurs de la dernière mesure récupérée sur le bus CAN.

La méthode « extraireMesures » reçoit en paramètre, les données de l’attribut « data » d’un objet de type « MCP_CAN » hérité de la classe « CAN », et traite les différents éléments de ce tableau pour les placer dans les attributs de la classe « Echantillon » (uPan, iPan...).

Cette méthode appelle la méthode « calculerCumulEnergie » afin de calculer l’énergie à partir des attributs de la classe.

La mesure se fait toutes les 30 minutes.

On fait l’hypothèse que la tension et l’intensité sont continues sur l’intervalle de mesure.

Formule littérale de l’énergie :

Formule littérale	$E = P \times t$
Unités adoptées par EDF/ERDF	E est exprimée en W.h
Puissance	$P = \text{Tension} \times \text{Intensité}$
Détails	E : énergie en wattheure (Wh) P : puissance en watt (W) t : temps en heure (h)

L’attribut « energie » de la classe Echantillon est calculée en Wh (Watt.heure).

Q17. Donner en C++ l'implémentation de la méthode « calculerCumulEnergie » permettant d'ajouter à l'attribut « energie » l'énergie produite depuis le dernier échantillon.

```
Void Echatillon::calculerCumulEnergie ()
{
    Energie=uPan*iPan*0.5+energie;
}
```

Q18. Compléter le code de la méthode « extraireMesures » permettant d'obtenir les mesures transmises par le bus CAN afin de créer un échantillon avant enregistrement dans la BDD. Consulter les **documentations PP1 et PP2.**

```
void Echantillon::extraireMesure(unsigned char *pMesures)
{
    // calcul de uPan
    unsigned short uInMSB=pMesures[0] & 0x03;
    unsigned short uInLSB=pMesures[1];
    unsigned short uIn=uInMSB<<8 | uInLSB;
    // 1023 est la valeur max possible sur 10 bits
    // 28V est la tension max
    uPan=((double)uIn/1023)*28;
```

```
    // calcul de iPan
    unsigned short iInMSB = pMesures[2] & 0x03 ;
    unsigned short iInLSB = pMesures[3] ;
    unsigned short iIn = iInMSB <<8 | iInLSB ;
    // 1023 est la valeur max possible sur 10 bits
    // 150A est l'intensite max
    iPan=((double) iIn / 1023) * 150 ;

    // calcul du cumul energetique
    calculerCumulEnergie() ;

    // calcul de uBat
    unsigned short uOutMSB = pMesures[4] & 0x03 ;
    unsigned short uOutLSB = pMesures[5] ;
    unsigned short uOut = uOutMSB <<8 | uOutLSB ;
    // 1023 est la valeur max possible sur 10 bits
    // 28V est la tension max
    uBat=((double) uOut/1023)*28;

    // calcul de la temperature
    temperature=(double) pMesures[6] ;
```

```
}
```

Les questions suivantes concernent la base de données et son évolution.

Consulter les documentations PP4 et PP5

La base de données « solaire » est constituée de deux tables.

La première sert à l'authentification des utilisateurs.

La seconde permet de stocker les différentes mesures.

Q19. Expliquer le rôle de la clé primaire « idMesure » de la table « mesures ».

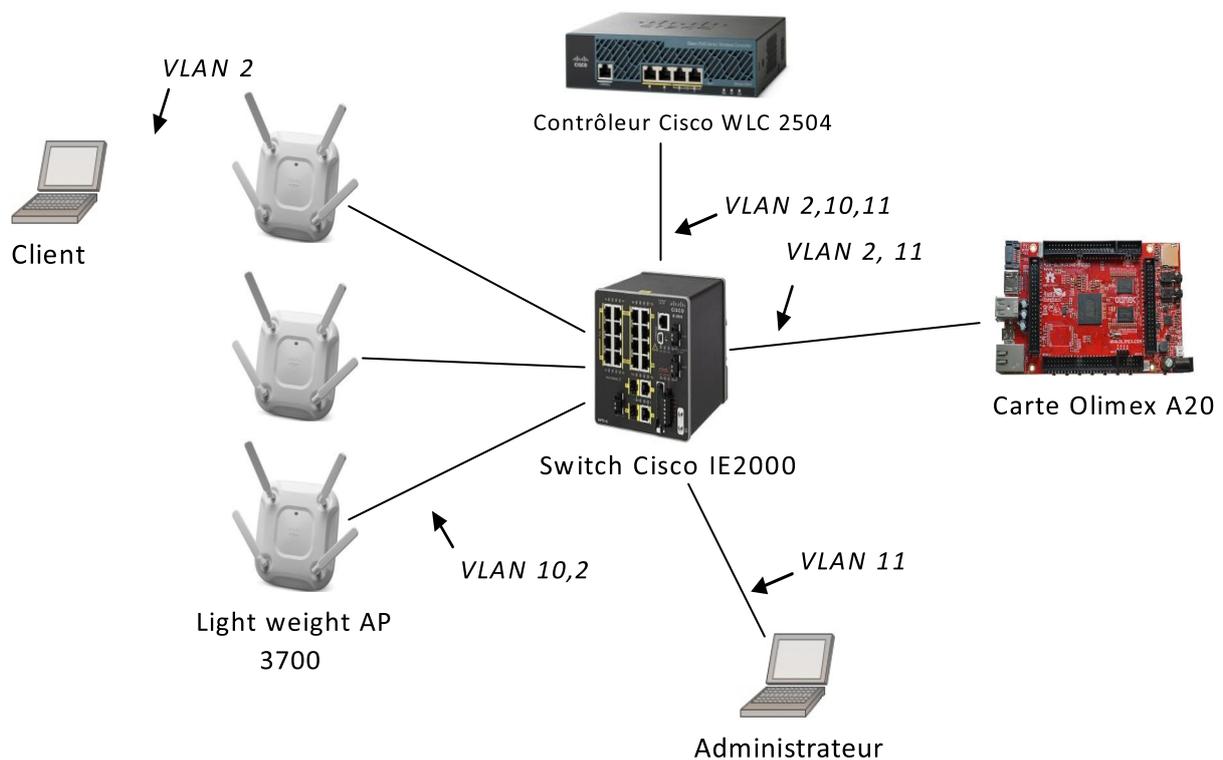
Une acquisition a donné pour résultats : uPan = 24.0, iPan = 18.8, energie = 321.2, uBat = 23.8, temperature = 34.0.

Le champ horodatage est affecté automatiquement lors de l'enregistrement dans la base de données.

Dans une base de données relationnelle, une clé primaire est une contrainte d'unicité qui permet d'identifier de manière unique un enregistrement dans une table.

Q20. Écrire en langage SQL la requête permettant d'écrire cette nouvelle acquisition dans la base de données.

```
INSERT INTO mesures(tension_pan, courant_pan, energie, tension_bat, temperature)
VALUES (24.0, 18.8, 321.2, 23.8, 34.0);
```



Partie E. ÉTUDE DU RÉSEAU

L'objectif de cette partie est de valider l'architecture du réseau.

Un logiciel de simulation du déploiement des points d'accès conseille l'implantation de trois points d'accès par wagon. Cette étude portera sur le wagon central.

La carte Olimex héberge le site Web à diffuser.

Les LAP (Lightweight Access Point) diffusent le réseau WIFI. Ils reçoivent leur configuration du contrôleur et diffusent le réseau WIFI.

Le contrôleur de réseau WIFI contrôle les différents LAP installés dans le wagon. Au démarrage du réseau, le contrôleur envoie la configuration aux LAP (puissance du signal, canal d'émission, sécurité, SSID...). Lorsqu'un client est connecté à un LAP, l'ensemble des messages en provenance ou à destination des clients est encapsulé par le LAP dans une trame CAPWAP à destination du contrôleur. Le contrôleur désencapsule ces trames et renvoie lesdites trames au récepteur.

Le VLAN 10 transporte les informations de configuration entre les LAP et le contrôleur WLC(réseau IP 192.168.10.0 /24).

Le VLAN 11 transporte les informations d'administration du réseau entre le PC administrateur et le contrôleur WLC (réseau IP 192.168.11.0 /24).Il est également possible depuis le PC Administrateur d'établir une connexion SSH vers la carte Olimex.

Le VLAN 2 transporte les données entre les clients et le serveur WEB (réseau IP 192.168.2.0 /24).

Remarque : certains équipements sont capables de communiquer sur plusieurs VLANs. Dans ce cas, ils possèdent une adresse IP dans chacun des VLANs auxquels ils sont associés.

Le site Web a été développé en langage PHP et est hébergé par le service apache sur la carte Olimex.

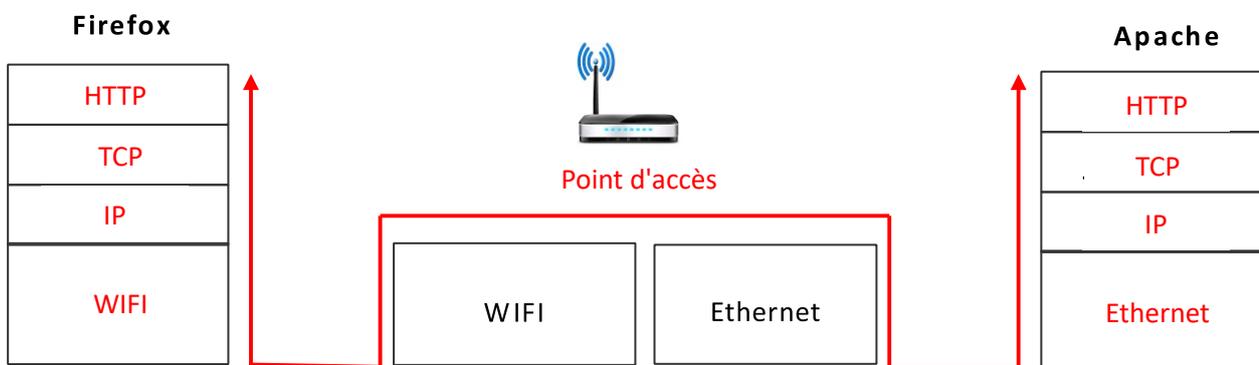
Q21. Compléter le schéma en utilisant les mots suivants : HTTP, TCP, IP, Ethernet, Point d'accès et WIFI.



PC portable



Carte Olimex



Q22. Dans le document réponses, renseigner les adresses IP et la valeur du masque (en décimal pointé) à donner à chaque équipement du réseau.

	VLAN 10 WLC 2504 – Com LAP ↔ WLC @MAC: 00 1B 54 93 62 20 @IP: 192.168.10.1 Mask: 255.255.255.0		VLAN 10 Access Point LAP1 @MAC: 00 1B 54 B3 97 64 @IP:192.168.10.2 Mask: 255.255.255.0
	VLAN 11 WLC 2504 – Com PC admin ↔ WLC @MAC: 00 1B 54 93 62 21 @IP: 192.168.11.2 Mask: 255.255.255.0		VLAN 10 Access Point LAP2 @MAC: 00 1B 54 A8 24 41 @IP: 192.168.10.3 Mask: 255.255.255.0
	VLAN 2 WLC 2504 – Com Serveur ↔ WLC @MAC: 00 1B 54 93 62 22 @IP: 192.168.2.10 Mask: 255.255.255.0		VLAN 10 Access Point LAP3 @MAC: 00 1B 54 12 D4 66 @IP: 192.168.11.3 Mask: 255.255.255.0

	VLAN 2 PC Client @MAC: 00 1B E9 78 96 FA @IP: 192.168.2.10 Mask: 255.255.255.0		VLAN 11 PC Administrateur @MAC: 00 1B E9 87 FE 21 @IP: 192.168.11.3 Mask: 255.255.255.0
	VLAN 2 Serveur Olimex (web) @MAC: 00 1B E9 41 23 65 @IP: 192.168.2.3 Mask: 255.255.255.0		VLAN 11 Serveur Olimex (ssh) @MAC: 00 1B E9 41 23 65 @IP: 192.168.11.4 Mask: 255.255.255.0

Afin d'optimiser le câblage réseau dans le train, on souhaite pouvoir acheminer plusieurs réseaux de niveau 2 sur un même support.

Q23. Indiquer la technologie qui permet cela.

Les VLANs

Q24. Indiquer comment les équipements connectés peuvent distinguer les trames appartenant à différents réseaux de niveau 2.

A l'aide de tags au niveau de la trame cf 802.1Q

Q25. Dans le document réponses, en vous aidant de la documentation PP6, décomposer la trame 802.1Q présente dans le document réponses en indiquant les adresses MAC source et destination, ainsi que les adresses IP source et destination, le type de protocole et le numéro de VLAN.

Trame 802.1Q (le préambule+SFD et le FCS ne sont pas présents)

```

offset                               data
0000      00 1b 54 93 62 20 00 1b 54 b3 97 64 81 00 00 0a
0010      08 00 45 00 00 ec 01 27 40 00 ff 11 e4 85 c0 a8
0020      0a 02 c0 a8 0a 01 e6 75 14 7f 00 d8 00 00 00 20
0030      03 20 00 00 00 00 01 04 d7 31 00 00 00 00 01 08
0040      2c 00 00 1b 54 b3 67 54 00 1b e9 78 96 fa 00 1b
0050      e9 41 23 65 81 00 00 02 08 00 aa aa 03 00 00 00
0060      08 00 45 00 00 a0 0f 00 40 00 80 06 65 96 c0 a8
0070      02 0a c0 a8 02 73 c3 58 1f 90 9d 1f 84 aa 11 53
0080      01 af 50 18 00 44 27 ca 00 00 47 45 54 20 68 74
0090      ... 2e 6d 73 66 74 6e 63 73 69 2e 63 6f 6d 2f 6e 63

00e0      48 6f 73 74 3a 20 77 77 77 2e 6d 73 66 74 6e 63
00f0      73 69 2e 63 6f 6d 0d 0a 0d 0a
  
```

Champ	Valeur
Adresse MAC destination	00 1b 53 93 62 20
Adresse MAC source	00 1b 54 b3 97 64
Numéro de VLAN	0000 0000 1010: VLAN10
Protocole de transport	11 → UDP
Adresse IP source	192.168.10.2
Adresse IP destination	192.168.10.1

Q26. Indiquer entre quels équipements se fait la communication précédente.

Entre le point d'accès LAP1 et le WLC

Q27. Indiquer si la valeur du champ VLAN ID est cohérente pour cet échange de trame. Justifier votre réponse.

Tag 0x81 00 00 0a → type de tag 802.1Q pour réseau Ethernet, donc, oui c'est cohérent (VLAN10).

DOCUMENTATION

Documentation PP1 : Caractéristiques techniques du MPPT

MPPT MECHANICAL SPECIFICATIONS

Parameter	Unit	Minimum	Typical	Maximum
Operating Temperature	°C	-30		70
Weight	g		2 650	
Length	mm		340	
Width	mm		190	
Height	mm		80	

MPPT ELECTRICAL SPECIFICATIONS

Parameter	Unit	Minimum	Typical	Maximum
Input Power Continuous	W	5		3240
Input Power Peak	W			3600
Input Current	A _{DC}			150
Peak Efficiency	%		97,5	
Input Voltage Range	V _{DC}		24	
Output Voltage Range	V _{DC}		24	

CAN ELECTRICAL SPECIFICATIONS

Supply Voltage	V _{DC}	6		18
SupplyCurrentRecessive	mA	15		50
SupplyCurrent Dominant	mA	60		100
Transmission Rate	kbits/s		125	
Bus length	m			500

ÉCHELLES DE MESURES ET DE CONVERSIONS ANALOGIQUES / NUMERIQUES

	U_{IN}		I_{IN}		U_{OUT}		$T_{AMBIANTE}$	
	Valeur codée sur 10 bits	Tension correspondante	Valeur codée sur 10 bits	Intensité correspondante	Valeur codée sur 10 bits	Tension correspondante	Valeur codée sur 8 bits	Température correspondante
min	0	0 V	0	0 A	0	0 V	- 128	-128°C
max	1023	28 V	1023	150 A	1023	28 V	127	127°C

Documentation PP2 Documentation bus CAN Présentation bus CAN

Le bus CAN est un réseau de terrain de type multi-maître dont le débit maximum est de 1 Mbit/s. Le procédé d'attribution du bus est basé sur le principe de l'arbitrage bit à bit, selon lequel les nœuds (ou stations) en compétition émettant simultanément sur le bus comparent bit à bit l'identificateur de leur message avec celui des messages concurrents.

Les stations sont câblées sur le bus par le principe du « ET câblé », et en cas de conflit, c'est à dire émission simultanée, la valeur 0 écrase la valeur 1.

L'état logique **0** est appelé état **dominant**.

L'état logique **1** est appelé état **récessif**.

Dès qu'une station émettant un état récessif détecte qu'une autre émet au même moment un état dominant, elle s'arrête d'émettre. Tous les perdants deviennent automatiquement des récepteurs du message et ne tentent à nouveau d'émettre que lorsque le bus se libère.

Le bus CAN implémente deux types de trames pour la communication entre les stations :

- Les **trames de requête** caractérisées par :
 - La valeur du champ RTR est un bit récessif
 - Champ DLC (Data Length Code) codé sur 4 bits indiquant le nombre d'octets de données attendus.
 - Champ Data vide.
- Les **trames de données** qui sont caractérisées par :
 - La valeur du champ RTR est un bit dominant
 - Champ DLC (Data Length Code) codé sur 4 bits indiquant le nombre d'octets de données du champ Data.

Valeurs pour les autres champs des trames CAN :

- SOF : 1 bit dominant
- Res : 2 bits dominants
- ACK : 2 bits récessifs
- EOF : 7 bits récessifs

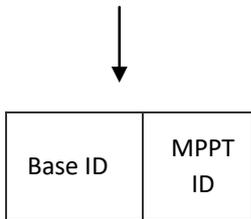
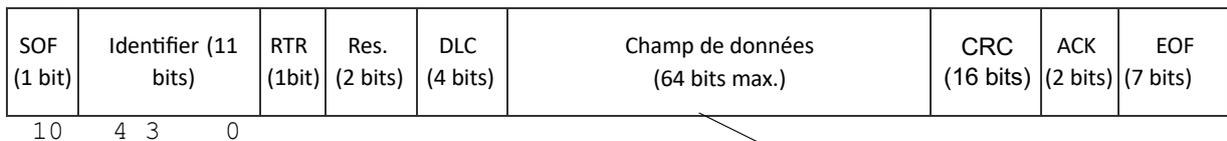
Caractéristiques des trames CAN (MPPT)

The CAN-driver circuit has to be powered externally.

CAN Communication Features :

- Full CAN V2.0A
- ID standard
- Nominal Bit rate 125 kbit/s
- Bus terminated on MPPT (124 Ohm)

The type of message used in this application is a standard data frame. Data frames consist of Arbitration Fields, Control Fields, Data Fields, CRC Fields, a 2 bit Acknowledge Field and an End of Frame.



Bits	7	6	5	4	3	2	1	0
Byte1	BVLR	OVT	NOC	UNDV	X	X	MSB U _{IN}	
Byte2	LSB U _{IN}							
Byte3	X	X	X	X	X	X	MSB I _{IN}	
Byte4	LSB I _{IN}							
Byte5	X	X	X	X	X	X	MSB U _{OUT}	
Byte6	LSB U _{OUT}							
Byte7	TAMBIANTE							

Bits 10-4 : Base ID

1110001 for master request frame

1110111 for MPPT answer frame

Bits 3-0 : MPPT ID Available range = 0001 to 1111

Note : All MPPT IDs must be different.

Bits 3 to 0 correspond to the DIP-switch setting on the MPPT.

BVLR : Battery Voltage Level Reached Flag 0 → U_{OUT} < U_{OUTMAX1} → U_{OUT} ≥ U_{OUTMAX}

OVT : Over Temperature Flag

0 → T_{COOLER} < T_{COOLMAX1} → T_{COOLER} ≥ T_{COOLMAX}

NOC : No Charge Flag

0 → Battery is connected 1 → Battery is not connected

UNDV : Under Voltage Flag 0

→ U_{IN} > 26V1 → U_{IN} ≤ 26V

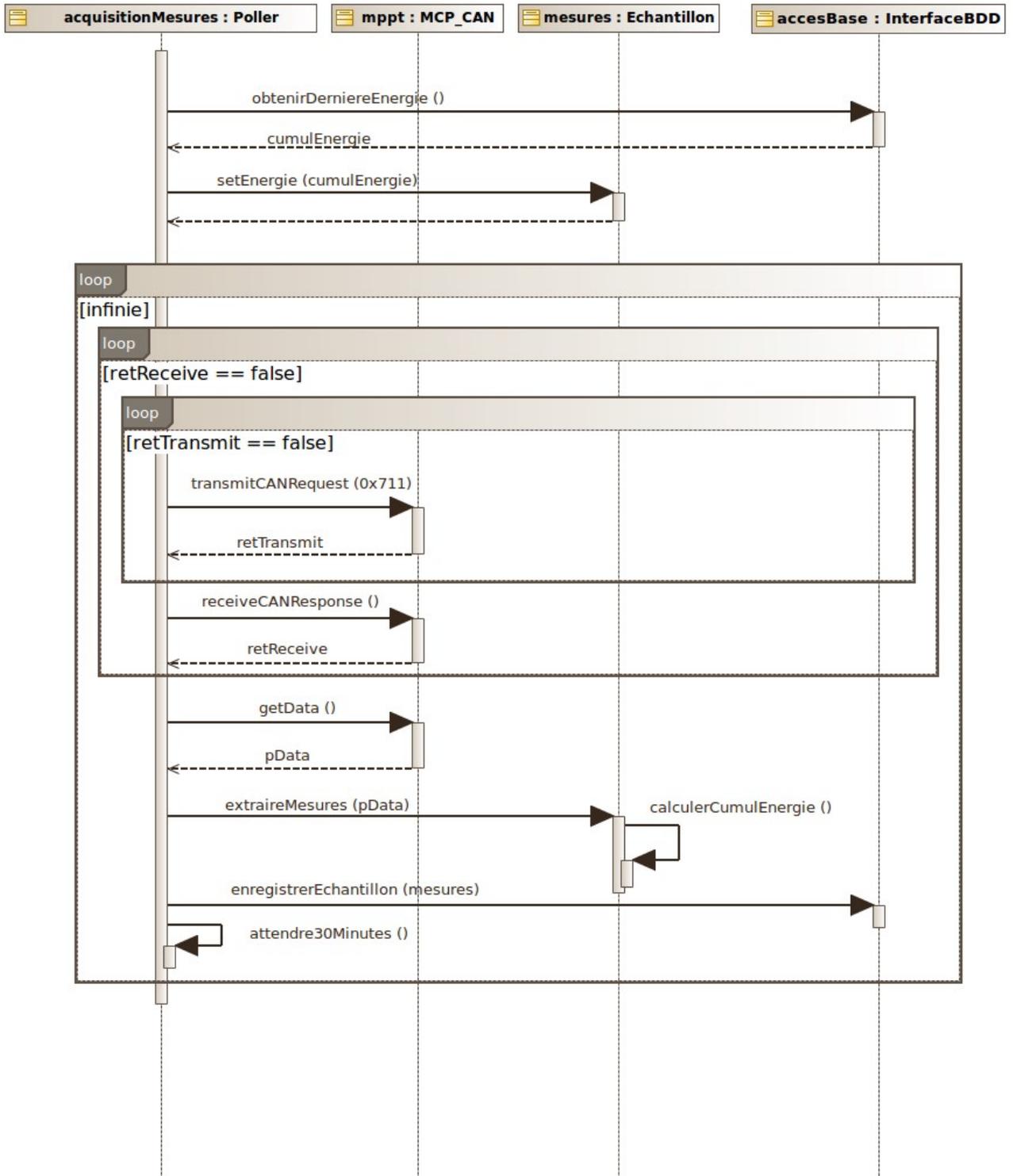
I_{IN}: 0 to 150 A U_{IN}:

0 to 28 V

U_{OUT}: 0 to 28 V

TAMBIANTE: Steps of 1°C

Documentation PP3 Diagramme de séquence



Documentation PP4 : Schéma relationnel de la base de données

La base de données est composée d'une table « mesures » et de quatre utilisateurs présents dans la table « utilisateurs ».

mesures	utilisateurs
<u>idMesure</u> horodatage tension_pan courant_pan energie tension_bat temperature	<u>idUser</u> login mdp

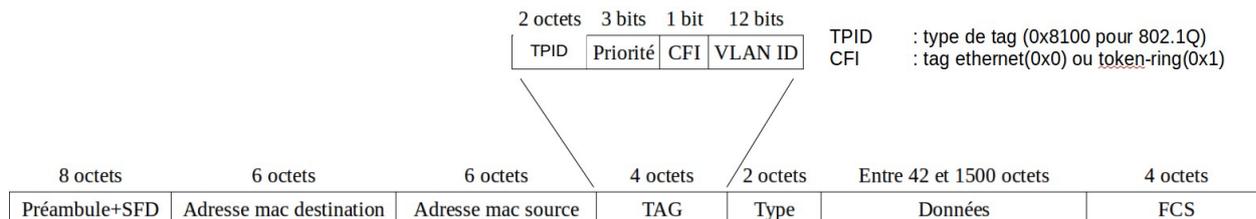
Documentation PP5 : Rappel des syntaxes SQL de Mysql

Utiliser (rendre active) une base de données existante :	USE nom_de_la_base;
Créer une base de données :	CREATE DATABASE nom_de_la_base;
Supprimer une base de données	DROP DATABASE nom_de_la_base;
Créer une table dans la base de données active:	CREATE TABLE nomTable (id INT NOT NULL AUTO_INCREMENT, champ1 DOUBLE, champ2 VARCHAR, champ3 TIMESTAMP NOT NULL, ..., PRIMARY KEY(id)) ;
Lister la structure d'une table :	DESCRIBE nomTable;
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 nomTable correspondant à deux critères.	SELECT * FROM nomTable WHERE nomChamp1 = 'poste' AND nomChamp3 < 12 ;
Sélectionner sur plusieurs tables nomTable1.nomChamp1 est clé primaire. nomTable2.nomChamp4 est une clé étrangère vers nomTable1.	SELECT * FROM nomTable1, nomTable2 WHERE nom_table1.nomChamp1 = nom_table2.nomChamp4 ;

Écrire une nouvelle entrée dans une table de BDD :	INSERT INTO nomTable(champ1, champ2) VALUES('valeur1', 'valeur2') ;
Modifier les informations de l'entrée dont le champ id = 51 :	UPDATE nomTable SET nomChamp1=10, valeur2=32 WHERE id=51 ;

Documentation PP6 Format de trame 802.1Q et paquet IP

Trame 802.1Q :



Paquet IP

