Étude d'un Système Numérique et d'Information
Option A (IR) - Session 2017
Correction tv (version 0.9)

Supervision d'une production d'énergie photovoltaïque

Partie B.

Q1.

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

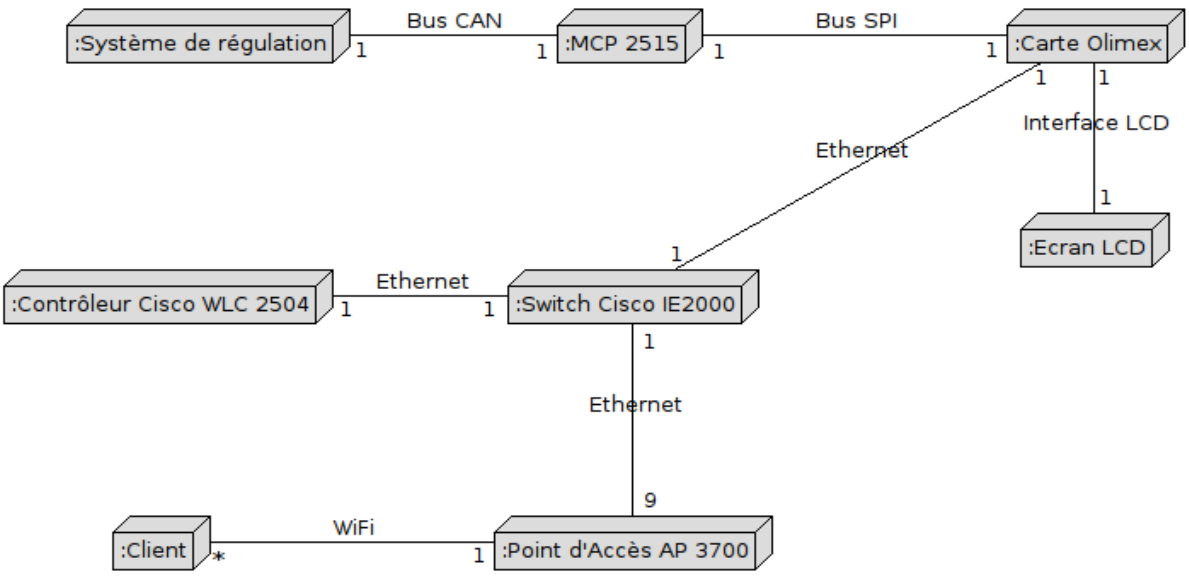
Q2. Justifier :

- la relation d'inclusion (« include ») : le cas d'utilisation « Acquérir les mesures ... » nécessite **obligatoirement** la réalisation du cas d'utilisation « Sauvegarder les échantillons ... » (comme c'est précisé dans l'analyse du contexte

Q3. Justifier :

- la relation d'extension (« extend ») : le cas d'utilisation « Lire les informations ... » **peut** être complété (**optionnellement**) par la réalisation du cas d 'utilisation « Générer un PDF »

Q4.



Partie C.

Q5. Rôle :

- SOF (*Start Of Frame*) : délimite le début de trame (permet au(x) récepteur(s) de se synchroniser)
- EOF (*End of Frame*) : délimite la fin de la trame (permet de libérer le bus)
- CRC (*Cyclic Redundancy Code*) : permet de détecter une erreur de transmission

Q6. Trame de requête (RTR = 1) ne contient pas de données :

SOF	**Identifier**	**RTR**	**Res**	**DLC**	**~~DATA~~**	**CRC**	**ACK**	**EOF**
0	11100010001	1	00	0111		….....	11	1111111
1 bit	11 bits	1 bit	2 bits	4 bits	0 bit	16 bits	2 bits	7 bits

Q7.

	Min.	Typ.	Max.
U_{IN}		24 V	
I_{IN}			150 A
U_{OUT}		24 V	
$T_{AMBIANTE}$	-30 °C		70 °C

Q8.

	Q	Valeurs hexa	Valeurs décimales
U_{IN}	28 / 1023 = 0,027370 V	11 0110 1101 = **0x36D**	**877** x Q = **24 V**
I_{IN}	150 / 1023 = 0,1466 A	00 1000 1111 = **0x08F**	**143** x Q = **20,967 A**
U_{OUT}	28 / 1023 = 0,027370 V	11 0110 1101 = **0x36D**	**877** x Q = **24 V**
$T_{AMBIANTE}$	1 °C	0001 1001 = **0x19**	**25 °C**

Partie D.

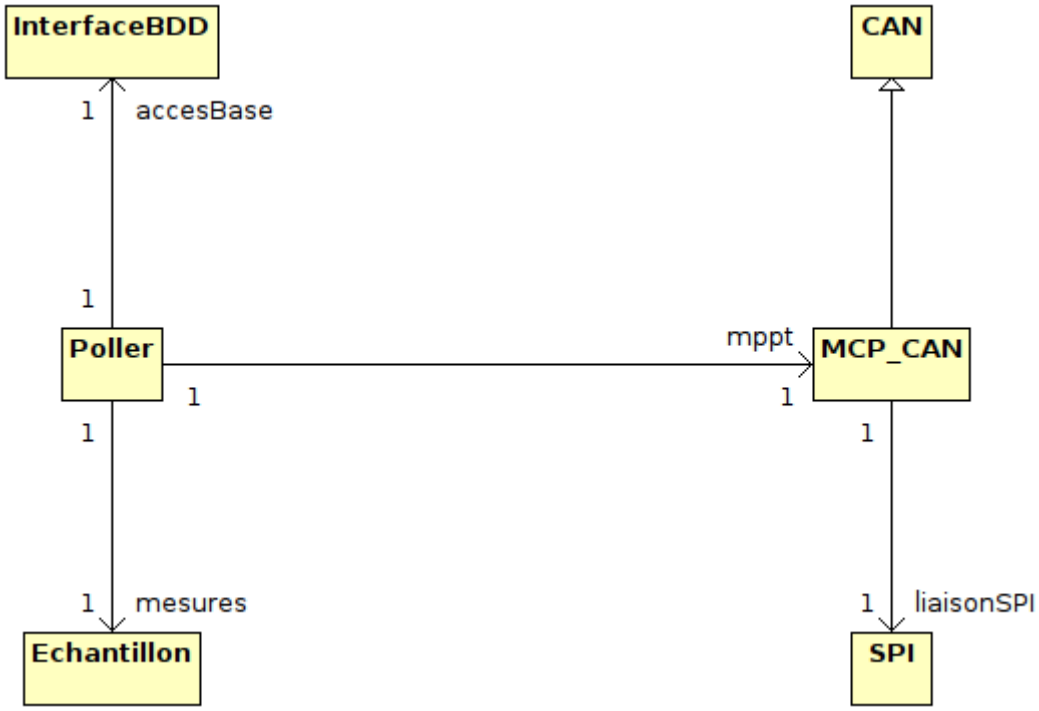
Q9. Relation :

- C'est une relation d'**héritage** : MCP_CAN hérite de CAN.
- CAN est une classe **générale** de gestion du bus CAN. MCP_CAN est une classe **spécialisée** pour la carte MCP2515 (qui est un contrôleur CAN d'où l'héritage) du système de supervision.

Q10.

Oui une instance de MCP_CAN peut accéder à l'attribut « data » de la classe CAN car « data » est un attribut protégé (**`protected`**) : il est donc accessible à la classe CAN et, par héritage, à tous ses enfants ici la classe MCP_CAN.

Q11. Compléter les associations (en fait ce sont des **compositions** qui sont implémentées dans le document !), les noms des rôles et les cardinalités :



Q12.

	Classe	Méthode
Envoyer une requête au MPTT	**`MCP_CAN`**	**`transmitCANRequest()`**

Q13.

	Classe	Méthode
Enregistrer un échantillon ...	**`InterfaceBDD`**	**`enregistrerEchantillon()`**

Q14.

	Classe	Méthode
Extraire les mesures ...	**`Echantillon`**	**`extraireMesures()`**

Q15. Constructeur de la classe **`Poller`** :

```
Poller::Poller
{
    // ...
    while(true)
    {
        bool retReceive = false;
        bool retTransmit = false;
        unsigned char *pData = NULL;

        while(retReceive == false)
        {
            while(retTransmit == false)
            {
                retTransmit = mppt.transmitCANRequest(0x711);
            }

            retReceive = mppt.receiveCANResponse();
        }

        pData = mppt.getData();

        mesures.extraireMesures(pData);

        accesBase.enregistrerEchantillon(mesures);

        attendre30Minutes();
    }
}
```

Q16.

```
class MCP_CAN : public CAN
{
    private:
        SPI liaisonSPI;

    public:
        bool receiveCANResponse();
        bool transmitCANRequest(unsigned int adr);
        MCP_CAN(unsigned int idCAN, unsigned int modeSPI, unsigned int
                vitesseSPI, unsigned int bitsParMotSPI);
};
```

Q17.

```
void Echantillon::calculerCumulEnergie()
{
    energie += (uPan * iPan) * 0.5; // U x I toutes les 30 minutes
}
```

Q18.

```
void Echantillon::extraireMesure(unsigned char *pMesures)
{
    unsigned short uInMSB = pMesures[0] & 0x03;
    unsigned short uInLSB = pMesures[1];
    unsigned short uIn = uInMSB << 8 | uInLSB;

    uPan = ((double)uIn / 1023) * 28;

    uInMSB = pMesures[2] & 0x03;
    uInLSB = pMesures[3];
    uIn = uInMSB << 8 | uInLSB;

    iPan = ((double)uIn / 1023) * 150;

    calculerCumulEnergie();

    uInMSB = pMesures[4] & 0x03;
    uInLSB = pMesures[5];
    uIn = uInMSB << 8 | uInLSB;

    uBat = ((double)uIn / 1023) * 28;

    temperature = (double)pMesures[6];
}
```

Q19. Clé primaire : **unicité** d'un enregistrement (*tuple*) dans la table **mesures**

Q20.

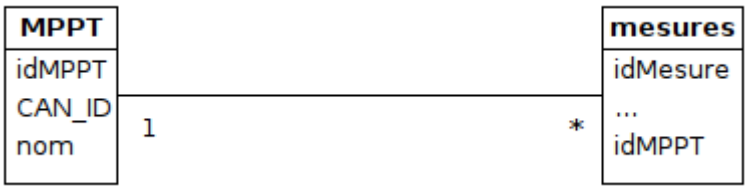
INSERT INTO mesures VALUES('', '', 24.0, 18.8, 321.2, 23.8, 34.0)

Q21.

```
CREATE TABLE `MPPT` (
  `idMPPT` int NOT NULL,
  `CAN_ID` int NOT NULL,
  `nom` varchar NOT NULL,
  PRIMARY KEY (`idMPPT`)
);
```

Q22. Le champ **idMPPT**

Q23.



Q24. **Clé étrangère**

Q25.

SELECT energie FROM mesures,MPPT
WHERE mesures.idMPPT = MPPT.idMPPT AND MPPT.CAN_ID = 3

Partie E. Réseau

Q26.

HTTP
TCP
IP
WiFi

Point d'Accès

WiFi	Ethernet

HTTP
TCP
IP
Ethernet

Q27.

VLAN10 **192.168.10.1/24**	**VLAN10** **192.168.10.2/24**
VLAN11 **192.168.11.1/24**	**VLAN10** **192.168.10.3/24**
VLAN2 **192.168.2.1/24**	**VLAN10** **192.168.10.4/24**
VLAN2 **192.168.2.2/24**	**VLAN11** **192.168.11.2/24**
VLAN2 **192.168.2.3/24**	**VLAN11** **192.168.11.3/24**

Q28. réseau local virtuel VLAN (*Virtual LAN*) par un commutateur (*switch*)

Q29.

- **VLAN de niveau 2 : par les adresses MAC**
- VLAN de niveau 3 : par les adresses IP

Q30.

Champ	Valeur
Adresse MAC destination	**00:1b:54:93:62:20**
Adresse MAC source	**00:1b:54:b3:97:64**
Numéro de VLAN	**0x000A (000 0 <u>000000001010</u>) = 10**
Protocole de transport	**0x11 = UDP**
Adresse IP source	**C0 A8 0A 02 = 192.168.10.2**
Adresse IP destination	**C0 A8 0A 01 = 192.168.10.1**

Q31.

192.168.10.2 AccessPoint LAP1	→	**192.168.10.1** WLC 2504

Q32. **Oui** les 2 équipements font partie du **VLAN 10** qui transporte les informations entre les LAP et le contrôleur WLC (réseau 192.168.10.0/24).

Q33.

- Le service **DHCP** (*Dynamic Host Configuration Protocol*) qui permet de configurer automatiquement l'adressage IP d'un hôte en lui affectant au moins une adresse IP et un masque de sous-réseau.

- Le service **DNS** (*Domain Name System*) qui permet de traduire un nom de domaine en adresses IP notamment.