

# DOCUMENTATION

Documentation PP1 : Protocole SDI-12	Page DOC2
Documentation PP2 : SDI-12 Reference Version 1.4	Page DOC3
Documentation PP3 : TEKBOX TBS06	Page DOC6
Documentation PP4 : Capteurs de niveau d'eau	Page DOC7
Documentation PP5 : Sonde de pression OTT-PLS	Page DOC8
Documentation PP6 : Éléments de codage	Page DOC10
Documentation PP7 : Langage PHP	Page DOC12
Documentation PP8 : Base de données « vigicrues »	Page DOC13
Documentation PP9 : Rappel des syntaxes SQL de Mysql	Page DOC14
Documentation PP10 : Adressage IP de la DREAL	Page DOC15
Documentation SP1 : Tableau comparatif technologies LP-WAN	Page DOC16

Session 2018	BTS Systèmes Numériques Option A Informatique et Réseaux Épreuve E4	Page DOC1 sur 16
18SN4SNIR1	Documentation	

# Documentation PP1 : Protocole SDI-12

---

Sources : <https://en.wikipedia.org/wiki/SDI-12>

## Généralités

SDI-12 (Serial Digital Interface à 1200 bauds) est un protocole de communication série asynchrone pour les capteurs intelligents qui surveillent les données environnementales. Ces capteurs sont utilisés dans des sites distants et communiquent habituellement avec un enregistreur de données ou un autre dispositif d'acquisition de données. Le protocole suit une configuration maître-esclave, grâce à quoi un enregistreur de données (*data logger*) demande les données des capteurs intelligents (*sensors*), identifiés chacun avec une adresse unique.

La communication s'effectue sur une seule ligne de données en *half-duplex*. Le système d'adressage numérique permet à un enregistreur de données de communiquer avec un maximum de 62 capteurs. Seul le capteur préconfiguré correspondant à cette adresse répondra. Les autres capteurs de la même ligne ne répondent pas tant qu'ils ne sont pas appelés et restent généralement en mode veille (*low power mode*).

Électriquement, le protocole est une connexion numérique à trois fils : données, masse et 12 V. Le signal de données, utilisant des niveaux logiques 5 V, est similaire à celui d'une liaison RS-232 avec le même codage d'octet asynchrone.

Toutes les communications SDI-12 sont transmises avec le codage ASCII à 1200 bauds avec 7 bits de données. La norme spécifie également un protocole de communication qui permet aux capteurs de rester dans un état de repos à faible consommation jusqu'à ce qu'ils soient réveillés par un signal « *break* » envoyé par le maître.

Le premier caractère de chaque commande est une adresse de capteur unique qui spécifie avec quel capteur l'enregistreur veut communiquer.

Session 2018	BTS Systèmes Numériques Option A Informatique et Réseaux Épreuve E4	Page DOC2 sur 16
18SN4SNIR1	Documentation	

# Documentation PP2 : SDI-12 Reference Version 1.4

Sources : <http://www.sdi-12.org/>

## SDI-12 ELECTRICAL INTERFACE

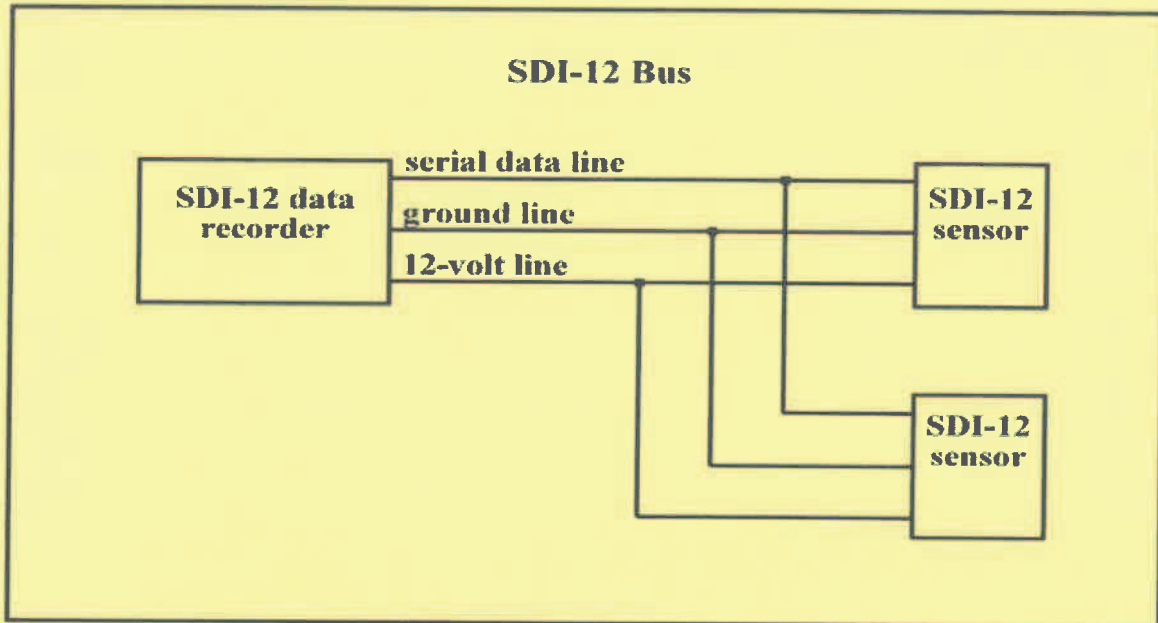


Figure 1. The SDI-12 Bus

A typical recorder/sensor measurement sequence proceeds as follows:

**Step 1.** The data recorder wakes all sensors on the SDI-12 bus.

**Step 2.** The recorder transmits a command to a specific, addressed sensor, instructing it to make a measurement.

**Step 3.** The addressed sensor responds within 15.0 milliseconds returning the maximum time until the measurement data will be ready and the number of data values it will return.

**Step 4.** If the measurement is immediately available, the recorder transmits a command to the sensor instructing it to return the measurement(s). If the measurement is not ready, the data recorder waits for the sensor to send a request to the recorder, which indicates that the data are ready. The recorder then transmits a command to get the data.

**Step 5.** The sensor responds, returning one or more measurements.

Session 2018	BTS Systèmes Numériques Option A Informatique et Réseaux Épreuve E4	Page DOC3 sur 16
18SN4SNIR1	Documentation	

## Baud Rate and Byte Frame Format

The baud rate for SDI-12 is 1200. Table 1 shows the byte frame format for SDI-12.

---

1 start bit
7 data bits, least significant bit transmitted first
1 parity bit, even parity
1 stop bit

---

*Table 1. SDI-12 byte frame format*

## Device Addresses

The first character of every command must be a sensor address. Likewise, the first character of a response is also the address character. This lets an SDI-12 recorder verify that the response has come from the correct sensor. (An address is a single character used to indicate which sensor is to respond to the command.) Table 2 shows the address codes.

ASCII address (a single character)	Decimal	Hex	Description
"0" (zero)	48	30	Default address, all sensors are initially set to "0" (zero) by the manufacturer for use in single sensor systems
"1" to "9"	49 to 57	31 to 39	Addresses for additional sensors on the SDI-12 bus

*Table 2. Sensor address codes*

ASCII '0' through ASCII '9' are the standard addresses which all sensors and data recorders must support. Should there be a need for more than 10 sensors, use an address in the range ASCII 'A' through ASCII 'Z' (decimal 65 through 90) and ASCII 'a' through ASCII 'z' (decimal 97 through 122).

Session 2018	BTS Systèmes Numériques Option A Informatique et Réseaux Épreuve E4	Page DOC4 sur 16
18SN4SNIR1	Documentation	

## Start Measurement Command (aM!)

This command tells the sensor to take a measurement. The sensor does not, however, return the measurement to the data recorder after this command. It returns the time until one or more measurements will be ready and the number of measurements that it will make. The send data (D0!) command must be issued to get the measurement(s). Table 3 shows the start measurement command.

Command	Response
aM!	attn<CR><LF>
a - the sensor address	a - the sensor address
M - the start measurement	ttt - the specified time, in seconds, until the sensor will have the measurement(s) ready
! - terminates the command	n - the number of measurement values the sensor will make and return in one or more subsequent D commands; n is a single digit integer with a valid range of 1 to 9

*Table 3. The start measurement command (aM!)*

If ttt is not equal to zero (000), the data recorder must wait for the specified time to elapse. Then it wakes the sensor and issues the D0 command.

When a data recorder issues an M command, it must complete the command/response sequence with the sensor before it sends any command to any other sensor. For example, suppose that following the issuance of the M command, the sensor responds

```
0M!      00101<CR><LF>
```

This response shows that one data value will be ready in 10 seconds.

## Examples of Start Measurement Command (aMC!) with CRC-16 and the Send Data Command (aD0!)

a. One measurement is immediately available after the MC command:

```
0MC!      00001<CR><LF>
```

```
0D0!      0+3.140qZ<CR><LF>
```

b. Three measurements will be ready 5 seconds after the MC command, and the sensor issues a service request. All 3 measurements are returned in response to the D0 command:

```
0MC!      00053<CR><LF>
```

```
0<CR><LF>
```

```
0D0!      0+3.14+2.718+1.414|pz<CR><LF>
```

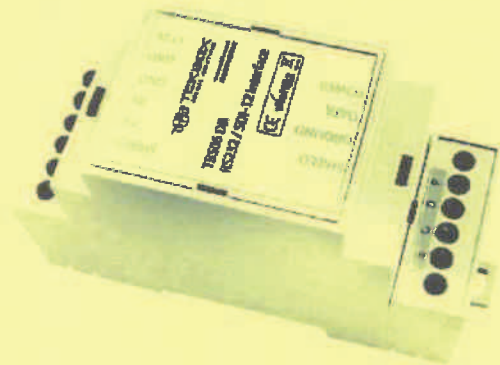
Session 2018	BTS Systèmes Numériques Option A Informatique et Réseaux Épreuve E4	Page DOC5 sur 16
18SN4SNIR1	Documentation	

# Documentation PP3 : TEKBOX TBS06

Sources : <https://www.tekbox.net/converters/sdi-12-converters/tbs06-sdi-12-to-rs232-converter>

## RS232 to SDI-12 Interface

The TBS06 (DR) RS232 to SDI-12 Converter is an interface for connecting a PC, data logger or RTU to one or more sensors with SDI-12 interface. The TBS06 connects to the RS232 interface and provides a SDI-12 compliant data interface. The TBS06 is a versatile tool for everyone who designs sensors and data recorders or who installs, tests or maintains SDI-12 based data acquisition systems.



## Overview

SDI-12 commands typically are ASCII strings generated by the data recorder/controller firmware. TBS06 converts the command strings to the logic levels and baud rate specified by the SDI-12 standard. Furthermore TBS06 handles breaks, marks and all other details of the SDI-12 protocol. Upon receiving data or status information originated by a Sensor, TBS06 extracts the corresponding ASCII strings and sends it to COM Port.

## Interface

The application is built based on a SDI-12 Recorder Protocol Stack Module. It receives commands from the RS232 Interface (e.g. via data logger, RTU or PC), and transfers the commands to the SDI Interface, waits for sensor response and transfers the response (measurement results, etc.) back to the RS232 Interface of the data logger, RTU or PC. All SDI-12 commands are supported. TBS06 supports RS232 or direct access to UART interface of the on board SDI-12 module.

## Product Features

- Plug and play. No driver required
- RS232 data rate: 4800 - 38400 Baud
- 6V ... 16V supply voltage
- Current consumption: 6 mA in "always on mode"; less than 250µA in "auto power management mode"

Session 2018	BTS Systèmes Numériques Option A Informatique et Réseaux Épreuve E4	Page DOC6 sur 16
18SN4SNIR1	Documentation	

# Documentation PP4 : Capteurs de niveau d'eau

## CAMPBELL SCIENTIFIC CS451

The CS451/CS456 Submersible Pressure Transducer provides depth of water, pressure and temperature measurements. It uses the SDI-12 or RS-232 communications protocols to communicate with an SDI-12 or RS-232 recorder simplifying installation and programming.



- Fully temperature compensated
- Power Requirements : 5 to 18 Vdc
- Power Consumption : Quiescent current <math>< 50 \mu\text{A}</math>, Measurement/Communication Current: 8 mA for 1-s measurement
- Measurement Time : Less than 1.5 s
- Outputs : SDI-12 (version 1.3) 1200 bps, RS-232 9600 bps
- Measurement Ranges : Pressure 0 to 1000 kPa, 0 to 102 m depth of fresh water

## OTT RLS - Radar Level Sensor

The OTT RLS is a non-contact radar level sensor with pulse radar technology. The OTT RLS offers a large measurement range with a small blanking distance and narrow beam width and it easily connects to most dataloggers. The RLS has extremely low power consumption and is ideal for remote or solar powered sites (requires only 12 mA @ 12V).



- Parameters measured : Water level / distance to water
- Measurement technology : Non-contact pulse radar
- Product highlights : Measures water level or depth to water from a bridge, pier or mounting arm
- Measurement range : 0.4 à 35 m
- Accuracy : 0.8 à 2 m:  $\pm 10\text{mm}$ ; 2 à 30 m:  $\pm 3\text{mm}$ ; 30 à 35 m:  $\pm 10\text{mm}$
- Interface : SDI-12, RS-485 (using SDI-12), or 4 ...20 mA

## PARATRONIC CR420/20

Designed to take measurements in natural environments, PARATRONIC radars are used to monitor the levels of rivers, channels, water bodies, or for sewage applications in pumping stations or storm outlets. The PARATRONIC CR420 or CRUZOE level sensors use pulse radar technology. Their very low power consumption enables them to be powered by batteries or solar panels.



- Measuring range : 20 meters
- Output signal : 4-20mA over 2 wires
- Power supply : 10 to 33 V DC
- Power : 22 mA during 2 seconds
- Accuracy : From 20cm to 50cm:  $\pm 30\text{mm}$ , From 50cm to 20m:  $\pm 5\text{mm}$

Session 2018	BTS Systèmes Numériques Option A Informatique et Réseaux Épreuve E4	Page DOC7 sur 16
18SN4SNIR1	Documentation	

# Documentation PP5 : Sonde de pression OTT-PLS

## Commandes et réponses SDI-12 standard (extraits du manuel d'utilisation)

Commande	Réponse	Description
<b>aI!</b>	<b>a13ccccccmmmmmm ... ... vvvxxxx&lt;CR&gt;&lt;LF&gt;</b>	Envoyer identification <b>a</b> - Adresse du capteur <b>13</b> - Version de protocole SDI-12 <b>ccccccc</b> - identification fabricant (nom de la société) <b>mmmmmm</b> - Désignation du capteur <b>vvv</b> - Version du capteur (Firmware) <b>xxxxxxx</b> - Numéro de série Réponse OTT PLS = 0130TTHACHPLS000100123456
<b>aAb!</b>	<b>b&lt;CR&gt;&lt;LF&gt;</b>	Modifier adresse du capteur <b>a</b> - Ancienne adresse du capteur <b>b</b> - Nouvelle adresse du capteur
<b>?!</b>	<b>a&lt;CR&gt;&lt;LF&gt;</b>	Demander adresse du capteur <b>a</b> - Adresse du capteur
<b>aM!</b>	<b>atttn&lt;CR&gt;&lt;LF&gt;</b> et au bout de 2 secondes <b>a&lt;CR&gt;&lt;LF&gt;</b>	Démarrer la mesure <b>a</b> - Adresse de capteur <b>ttn</b> - Temps en secondes jusqu'à ce que le capteur ait déterminé le résultat de mesure Réponse OTT PLS = 002 <b>n</b> - Nombre de mesures Réponse OTT PLS = 2 <b>a&lt;CR&gt;&lt;LF&gt;</b> - Requête de service
<b>aD0!</b>	<b>a&lt;valeur1&gt;&lt;valeur2&gt;&lt;CR&gt;&lt;LF&gt;</b>	Envoyer données <b>a</b> - Adresse du capteur <b>&lt;valeur1&gt;</b> - Valeur de niveau/de pression Formats de la mesure : m → pbbbb.eee cm → pbbbbbbb ft → pbbbb.ee mbar → pbbbbbb.e psi → pbbbb.eee <b>&lt;valeur2&gt;</b> - Valeur de température Formats de la mesure : °C et °F → pbbb.e <b>p</b> - Signe (+,-) <b>b</b> - Chiffres (avant la virgule) <b>e</b> - Chiffres après la virgule
<b>aMC!</b>	<b>atttn&lt;CR&gt;&lt;LF&gt;</b> et au bout de 2 secondes <b>a&lt;CR&gt;&lt;LF&gt;</b>	Démarrer la mesure et demander un CRC (Cyclic Redundancy Check) ; pour plus de détails voir commande <b>aM!</b> . La réponse à la commande suivante <b>aD0!</b> contient une valeur CRC en plus : <b>a&lt;valeur1&gt;&lt;valeur2&gt;&lt;CRC&gt;&lt;CR&gt;&lt;LF&gt;</b>



## Commandes et réponses SDI-12 étendues (extraits du manuel d'utilisation)

Toutes les commandes SDI-12 étendues commencent par un « O » comme OTT. Ces commandes permettent de configurer OTT PLS.

Commande	Réponse	Description
▶ Définir/Lire unité des valeurs mesurées de niveau/pression		
aOSU<valeur>!	a<valeur><CR><LF>	Définir unité
aOSU!	a<valeur><CR><LF>	Lire unité
<b>a</b> - Adresse du capteur		
<b>&lt;valeur&gt;</b> - <b>Unités pour la mesure du niveau</b>		
+0 = m; réglage d'usine		
+1 = cm		
+2 = ft		
La mesure du niveau est effectuée avec compensation de la densité de l'eau, de la température de l'eau et de l'accélération de la pesanteur locale !		
<b>Unités pour la mesure de la pression</b>		
+3 = mbar		
+4 = psi		
La mesure de la pression est effectuée sans compensation !		
<b>Attention</b>		
Si avant de modifier l'unité, des réglages ont déjà été effectués pour les paramètres "Offset", "Valeur de référence", "Seuil inférieur" ou "Seuil supérieur", ils doivent être réitérés ! Il n'y a pas de conversion automatique des paramètres définis !		
▶ Définir/Lire unité des valeurs mesurées de température		
aOST<valeur>!	a<valeur><CR><LF>	Définir unité
aOST!	a<valeur><CR><LF>	Lire unité
<b>a</b> - Adresse du capteur		
<b>&lt;valeur&gt;</b> - +0 = °C ; réglage d'usine		
+1 = °F		

# Documentation PP6 : Éléments de codage

Remarque : Ces extraits ne concernent que le codage des relations entre les classes

Fichier pqtTD\_User.h

```
class TD_USER {  
    private:  
        C_Pluie * ptrObjC_Pluie;  
        C_Niveau * ptrObjC_Niveau;  
        C_RS232 objC_RS232;  
        C_RadioSigfox * ptrObjC_RadioSigfox;  
    public:  
        TD_USER(void) ;  
        ~TD_USER(void) ;  
        void TD_USER_Setup(void) ;  
        void TD_USER_Loop(void) ;  
};
```

Fichier pqtTD\_User.cpp

```
#include "pqtTD_User.h"  
TD_USER :: TD_USER(void) {  
    ptrObjC_RadioSigfox = new C_RadioSigfox() ;  
    ptrObjC_Pluie = NULL ;  
    // ...  
}  
TD_USER :: ~TD_USER(void) {  
    delete ptrObjC_RadioSigfox ; // ...  
}  
void TD_USER :: TD_USER_Setup(void) {  
    ptrObjC_Pluie = new C_Pluie(&objC_RS232) ; // ...  
}  
void TD_USER :: TD_USER_Loop(void) { // ...  
}
```

Session 2018	BTS Systèmes Numériques Option A Informatique et Réseaux Épreuve E4	Page DOC10 sur 16
18SN4SNIR1	Documentation	

Fichier pqtCapteurs.h

```
#include "pqtSerie.h"

class C_SDI_12 {
protected:
    C_RS232 * ptrC_RS232;
    // ...
public :
    C_SDI_12(C_RS232 * prmRS232) ;
    ~C_SDI_12(void) ;
    // ...
private :
    // ...
};

class C_Pluie: public C_SDI_12 {
public :
    C_Pluie(C_RS232 * prmRS232) ;
    ~C_Pluie(void) ;
    // ...
};
```

Session 2018	BTS Systèmes Numériques Option A Informatique et Réseaux Épreuve E4	Page DOC11 sur 16
18SN4SNIR1	Documentation	

# Documentation PP7 : Langage PHP

---

## Les bases

Le langage PHP est un langage de programmation interprété qui s'exécute sur un serveur Web.

Chaque instruction se termine par un point-virgule « ; » comme en C. Les commentaires s'écrivent de la même façon qu'en langage C.

Les variables sont typées à l'exécution suivant le contexte dans lequel la variable est utilisée. Toutes les variables sont préfixées par le symbole « \$ ». Par exemple :

```
$foo = "essai" ; // la variable $foo contient alors une chaîne de caractères (string)
```

```
$foo = 12.5 ; // la variable $foo contient alors un nombre à virgule (float)
```

Les types PHP sont : string, bool, int, float, NULL. On peut également créer des tableaux.

L'interpréteur PHP met à disposition un ensemble de fonctions utilisables comme des fonctions statiques C.

## Fonction substr() : Retourne un segment de chaîne

```
string substr ( string $string , int $start [, int $length ] )
```

- Valeurs de retour : Retourne le segment de \$string défini par \$start et \$length.
- \$string : La chaîne d'entrée. Doit comporter au moins un caractère.
- \$start : La chaîne retournée commencera au caractère numéro \$start, dans la chaîne \$string. Le premier caractère est numéroté zéro.
- \$length : La chaîne retournée contiendra au plus \$length caractères, en commençant à partir du caractère \$start

Exemple :

```
$result = substr("abcdef", 1, 3); // la variable $result contiendra la chaîne "bcd"
```

## Fonction hexdec() : Convertit de hexadécimal en décimal

```
number hexdec ( string $hex_string )
```

- Valeurs de retour : Retourne un nombre entier correspondant à la représentation hexadécimale contenue dans la chaîne \$hex\_string.
- \$hex\_string : La chaîne hexadécimale à convertir

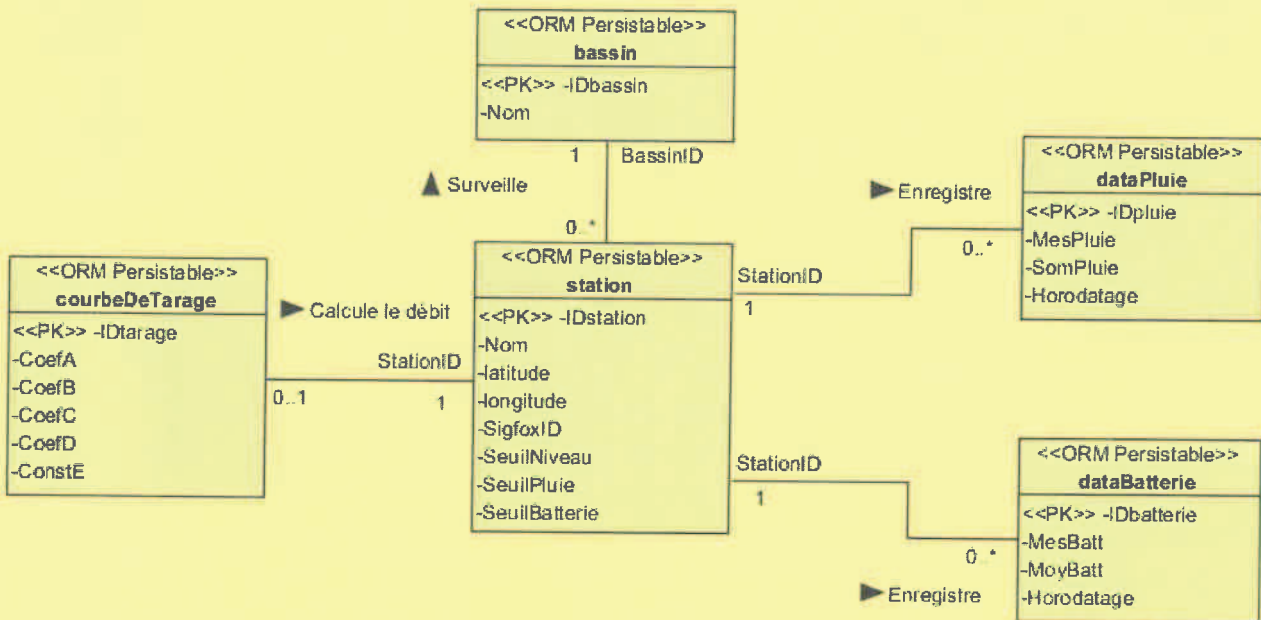
Exemple :

```
$result = hexdec("a0") ; // la variable $result contiendra le nombre entier 160
```

Session 2018	BTS Systèmes Numériques Option A Informatique et Réseaux Épreuve E4	Page DOC12 sur 16
18SN4SNIR1	Documentation	

# Documentation PP8 : Base de données « vigicrues »

Le service web de collecte enregistre les données des stations de mesures dans la base de données MySQL « vigicrues ». Cette base de données est décrite par le schéma conceptuel suivant (notation UML2) :



Ce modèle conceptuel générera le code SQL suivant pour créer la table dataBatterie :

```
CREATE TABLE databatterie (
  IDbatterie int(10) NOT NULL AUTO_INCREMENT,
  StationID int(10) NOT NULL,
  MesBatt float DEFAULT NULL,
  MoyBatt float DEFAULT NULL,
  Horodatage timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (IDbatterie),
  CONSTRAINT FOREIGN KEY (StationID) REFERENCES station
(IDstation)
);
```

Le champ Horodatage des tables est rempli automatiquement par Mysql avec la date courante lors de chaque insertion d'un nouvel enregistrement.

La table courbeDeTarage contient les coefficients de la fonction polynomiale d'ordre 4 qui interpole la courbe de tarage établie par l'hydrologue.

Débit = CoefA \* Niveau<sup>4</sup> + CoefB \* Niveau<sup>3</sup> + CoefC \* Niveau<sup>2</sup> + CoefD \* Niveau + ConstE

Session 2018	BTS Systèmes Numériques Option A Informatique et Réseaux Épreuve E4	Page DOC13 sur 16
18SN4SNIR1	Documentation	

# Documentation PP9 : Rappel des syntaxes SQL de Mysql

<b>Utiliser</b> (rendre active) une base de données existante :	USE nom_de_la_base ;
<b>Créer</b> une base de données :	CREATE DATABASE nom_de_la_base ;
<b>Supprimer</b> une base de données	DROP DATABASE nom_de_la_base ;
<b>Créer</b> 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)) ;
<b>Sélectionner</b> toutes les informations de la table :	SELECT * FROM nomTable ;
<b>Sélectionner</b> seulement les informations d'un champ :	SELECT nomChamp FROM nomTable ;
<b>Sélectionner</b> tous les champs de la table nomTable correspondant à deux critères.	SELECT * FROM nomTable WHERE nomChamp1 = 'poste' AND nomChamp3 < 12 ;
<b>Sélectionner</b> 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 ;
<b>Trier les résultats</b> par ordre croissant	SELECT nomChamp FROM nomTable ORDER BY nomChamp ASC ;
<b>Trier les résultats</b> par ordre décroissant	SELECT nomChamp FROM nomTable ORDER BY nomChamp DESC ;
<b>Écrire</b> une nouvelle entrée dans une table de BDD :	INSERT INTO nomTable(champ1, champ2) VALUES('valeur_textel', 0.867) ;
<b>Modifier</b> les informations de l'entrée dont le champ id = 51 :	UPDATE nomTable SET nomChamp1=10, valeur2=32 WHERE id=51 ;
<b>Supprimer</b> une entrée dans une table	DELETE FROM nomTable WHERE champ1='valeur1' ;

Session 2018	BTS Systèmes Numériques Option A Informatique et Réseaux Épreuve E4	Page DOC14 sur 16
18SN4SNIR1	Documentation	

# Documentation PP10 : Adressage IP de la DREAL

## Adresses de quelques éléments du réseau

Réseau	Élément réseau	Adresse IP + masque	Adresse MAC	Passerelle par défaut
Internet	R1 WAN	37.235.89.249 /30		37.235.89.250
Dorsale	R1 LAN	172.16.240.1 /26	00:2A:10:D4:FB:11	-
	R2 Eth1	172.16.240.2 /26	00 :2A :10 :E0 :A5 :21	172.16.240.1
	R3 Eth1	172.16.240.3 /26	00 :2A :10 :B3 :E1 :31	172.16.240.1
	R4 Eth1	172.16.240.4 /26	00 :2A :10 :B2 :FB :41	172.16.240.1
SPC	R2 Eth2	172.16.0.1 /20	00 :2A :10 :E0 :A5 :22	-
DMZ	R1 DMZ	172.16.240.65 /26	00:2A:10:D4:FB:13	-
	Service web de collecte	172.16.240.66 /26	00:2A:10:A7:AA:59	172.16.240.65
	Site web de consultation	172.16.240.67 /26	00:2A:10:A7:BF:62	172.16.240.65

Session 2018	BTS Systèmes Numériques Option A Informatique et Réseaux Épreuve E4	Page DOC15 sur 16
18SN4SNIR1	Documentation	

# Documentation SP1 : Tableau comparatif technologies LP-WAN

LP-WAN : Low Power Wide Area Network

Fonctionnalité	LORA	Sigfox	LTE M (4G)
Modulation	SSC	UNB DBPSK – GFSK ...	OFDMA
Bande passante	500 Hz - 125 kHz	100Hz	1,4 MHz
Débit de données	290 bps à 50 kbps	100 bps	200 kps à 1 Mbps
Message par jour	illimité	140	illimité
Puissance d'émission	14 dBm	14dBm	20 dBm
Bilan de liaison	154 dB	151 dB	146 dB
Portée en zone urbaine	1 km	3 à 10 km	
Portée en zone rurale	10 km	30 km	
Durée de vie de batterie 2 Ah	105 mois	90 mois	19 mois
Efficacité énergétique	très élevée	très élevée	moyenne

Source : [www.vertical-m2m.com.fr](http://www.vertical-m2m.com.fr)

<https://fr.slideshare.net/RobertVivancoSalcedo/comprendre-les-technologies-lpwa-sigfox-et-lora>

Session 2018	BTS Systèmes Numériques Option A Informatique et Réseaux Épreuve E4	Page DOC16 sur 16
18SN4SNIR1	Documentation	