Station météo – Cahier des charges

**Table des matières**

[1. Description 3](#_Toc102505659)

[2. Communication 3](#_Toc102505660)

[2.1. Connexion 4](#_Toc102505661)

[2.2. Contenu 4](#_Toc102505662)

[2.2.1. Contenu de trame 4](#_Toc102505663)

[2.2.2. Mesures 5](#_Toc102505664)

[2.2.3. Alarmes CAPTEUR 7](#_Toc102505665)

[3. L’interface graphique 7](#_Toc102505666)

[3.1. Tableau de mesures 7](#_Toc102505667)

[3.2. Tableau d’alarmes 7](#_Toc102505668)

[3.3. Affichage de mesures en graphique 8](#_Toc102505669)

[3.4. Gestion de comptes utilisateurs 8](#_Toc102505670)

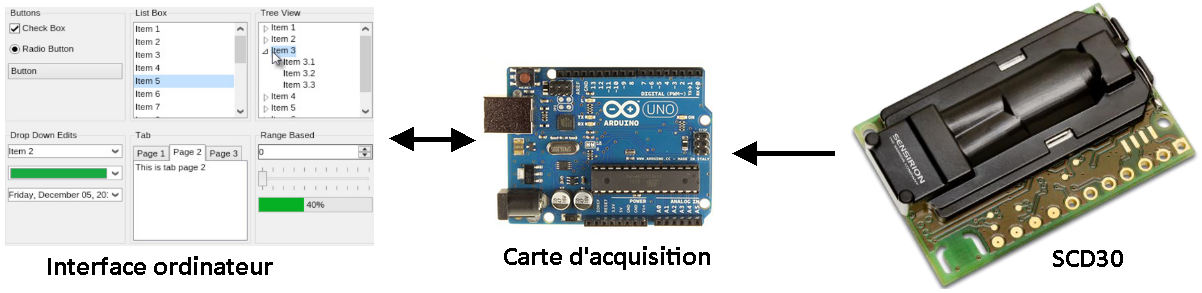
[4. La sauvegarde en fichiers 9](#_Toc102505671)

[5. La sauvegarde en base de données 9](#_Toc102505672)

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Modifications** | **Version software** |
| 1.0 | 07/02/2022 | Première version du cahier des charges | Draft |
| 1.1 | 14/02/2022 | Création de l’objet « mesure » :  2.2 complété | Draft |
| 2.0 | 21/02/2022 | Revue globale du document suivant exigences client | Draft |
| 2.1 | 01/03/2022 | 2.2.1 : proposition de contenu de trame  2.2.1.5 : Définition du checksum | **1.0** :  - Acquisition de l’information venant des capteurs suivant prédéfinition de trame (2.2)  - Affichage des données reçues en Datagrid (3.1)  - Gestion de sélection des 6 pages graphiques en UserControl |
| 2.2 | 14/03/2022 | 3.3 mis à jour  2.2 globalement adapté au changement de contenu trame suite à réunion IG-EO du 15/03/2022 | **1.1 :**  - Affichage des alarmes UTILISATEUR en Datagrid (3.2)  - Affichage des mesures en graphique (3.3)  - Création de compte utilisateur (3.4) |
| 3.0 | 18/04/2022 | 4 mis à jour | **2.0 :**  - Sauvegarde de paramètres en fichier |
| 4.0 | 03/05/2022 | 5 mis à jour | **3.0 :**  - Sauvegarde des tables utilisateur en base de données ACCESS  - Sauvegarde des mesures en base de données ACCESS |

# Description

Le SCD30 est un module électronique capable de mesurer 3 paramètres environnementaux différents : la concentration en CO2 dans l’air, la température, l’humidité. Elle se présente sous la forme d’une carte équipée de capteurs à connecter sur un port USB d’ordinateur via une carte d’acquisition des données.



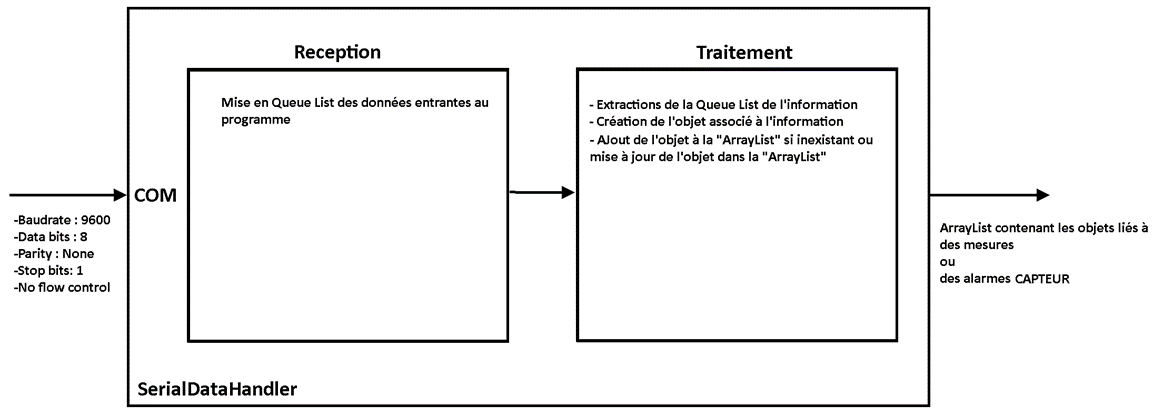
La carte d’acquisition envoie toutes les 30 secondes 3 mesures sur un port COM de l’ordinateur. Votre travail consiste à développer une interface ordinateur, en C#, capable de :

* Récolter les mesures, les stocker et les présenter en interface graphique
* Récolter les alarmes CAPTEUR entre carte d’acquisition et SCD30 (problème de connexion, entretient à prévoir,…)
* Permettre à l’utilisateur de gérer des alarmes UTILISATEUR de dépassement sur ces mesures
* Gérer la sauvegarde/chargement de paramètres de configuration en fichier CSV
* Gérer des comptes utilisateurs et l’octroi de droits d’accès
* Sauvegarder mesures et comptes utilisateurs en base de donnée ACCESS

L’interface doit être **évolutive**. Son développement doit tenir compte du fait que d’autres types de mesures peuvent dans le temps s’ajouter à la carte d’acquisition.

# Communication

La carte d’acquisition se connecte par USB à l’ordinateur. Elle apparaît dans l’interface sous forme de port d’entrée/sortie « COM ». L’objet « SerialPort » dans l’environnement .NET permet la prise en charge des informations venant d’un port COM. Le module software à développer doit correspondre au schéma suivant :



## Connexion

Les paramètres de connexion de la carte d’acquisition à l’ordinateur sont:

* Baudrate : 9600
* Data bits : 8
* Parity : None
* Stop bits: 1
* No flow control

Le port COM doit être trouvé automatiquement par l’interface.

## Contenu

Deux types de données peuvent être échangées entre carte d’acquisition et interface :

* Des mesures de concentration CO2, humidité, température
* Des alarmes capteurs (capteur défaillant, calibration de capteur,…)

### Contenu de trame

Une trame de données échangée entre carte d’acquisition et interface doit contenir au minimum :

* Un identifiant
* Le type de donnée échangée
* ~~La taille de la donnée échangée~~
* La donnée brut échangée

Voici la définition d’une trame de données contenant une mesure :

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Entête début  **0x55 0x55 0xAA** | Serial 1 | Serial 0 | ID | Type | Data1 | Data0 | Checksum | Entête Fin  **0xAA 0xAA 0x55** |

#### Le numéro de série « Serial »

Le numéro de série est propre à la carte d’acquisition et est unique. Celui-ci est codé sur 2 octets et indique le numéro de série de la carte d’acquisition.

#### L’identifiant « ID »

L’identifiant est généré par la carte d’acquisition et est sur 1 octet. Il prend une valeurs de 1 à 255 afin d’identifier chaque mesure, Il est donc unique par carte d’acquisition.

Ex : Le capteur de température dans le coin supérieur gauche avant de la pièce a l’ID 7.

La valeur 0 est réservée pour les alarmes CAPTEUR.

#### Type de donnée « TYPE »

Le type de données est transmis par la carte d’acquisition.

|  |  |
| --- | --- |
| **Ref.** | **TYPE** |
| 0 | Alarme |
| 1 | CO² |
| 2 | Température |
| 3 | Humidité |

#### ~~Taille de donnée échangée « NbrBytes »~~

~~La donnée brut est échangée sous format binaire, de 1 à 4 octets.~~

#### La donnée brut échangée « DATA »

La donnée envoyée par la carte d’acquisition est en format binaire, codée sur 2 octets. Sa valeur va de

* 0 pour la plus faible mesure réalisable par le capteur,
* à (2^16 - 1) = 65535 pour la plus haute mesure réalisable par le capteur.

#### Le checksum « CS »

Le checksum permet de détecter un défaut de transmission de l’information de l’émission à la réception. Le principe est le suivant :

La carte d’acquisition additionne les octets ID, TYPE, ~~NBRDATA~~, DATA et envoie le résultat sur 1 octet dans la trame.

L’interface locale réceptionne la trame, recalcule le checksum et le compare au checksum réceptionné, présent dans la même trame. Le checksum reçu par l’interface et le checksum calculé par l’interface doivent être identiques.

### Mesures

Pour la mesure, les paramètres de configuration suivants doivent être ajoutés :

* L’intervalle de mesure
* La donnée convertie
* Définition d’alarme de période de rafraichissement maximum
* Définition de seuils d’attention min. et max.
* Définition de seuils critiques min. et max.

Exemple :

|  |  |
| --- | --- |
| **humidité** | **Exemples de données/configurations** |
| Identifiant | 5 (de 1 à 200) |
| Type | 3 (4 types actuellement identifiés : Temp, humid, CO², alarmes) |
| Donnée brut | 30322 sur une plage de 0 à 65535 (donnée sur 16bits) |
| Intervalle de mesure | 0 à 100% |
| Donnée convertie | 46% sur une plage de 0 (0 binaire) à 100% (65535 binaire) |
| Alarme période | 10 sec. Activation de l’alarme si la donnée n’est pas rafraichie depuis plus de 10secondes. |
| Warning min. | 40% (0% si inactivé) |
| Warning max. | 70% (100% si inactivé) |
| Critical min. | 30% (0% si inactivé) |
| Critical max. | 80% (100% si inactivé) |

La donnée convertie est calculée sur base de la donnée brut contenue dans la trame suivant la formule :

Avec

n, nombre de bit de la donnée

MAX, valeur maximale de la donnée convertie

MIN, valeur minimale de la donnée convertie

Exemple : L’ID2 est une température codée sur 16bits.

* La mesure non convertie est dans l’intervalle [0 - 65535], par exemple 17620
* La mesure convertie est entre -40°C (0 non convertit) et 80°C (65535 non convertit)
* La mesure convertie est :

### Alarmes CAPTEUR

# L’interface graphique

## Tableau de mesures

L’interface doit permettre de visualiser les mesures envoyées par la carte d’acquisition ainsi que leurs paramètres de configuration. Voici un exemple d’affichage :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Config status** | **Type** | **Data** | **Last Update** | **Alarm** |
| 1 | Done | CO² | 560ppm | 10 seconds | Ok |
| 2 | Not done | Humidity | - | 40 seconds | - |
| 3 | Done | Temperature | 27°C | 60 seconds | High |

* **ID :** Indique l’ID de la mesure, attribué à la mesure par la carte d’acquisition.
* **Config Status :** Indique si l’ID a été configuré par l’utilisateur
* **Type :** Indique le type de mesure (CO², Temperature, Humidity)
* **Data :** Indique la donnée convertie, « - » si non ID configuré
* **Last Update :** Indique la durée en secondes depuis la dernière d’une trame de l’ID concerné
* **Alarm :** Indique l’état de l’alarme utilisateur (Too Low – Low – Ok – High – Too High)

## Tableau d’alarmes

L’interface doit permettre de visualiser les alarmes UTILISATEUR ainsi que leurs paramètres de configuration. Voici un exemple d’affichage :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Type** | **Critical Min** | **Warning Min** | **Warning Max** | **Critical Max** | **Status** |
| 1 | CO² | 0ppm | 0ppm | 700ppm | 1200ppm | Ok |
| 2 | Humidity | - | - | - | - | - |
| 3 | Temperature | 18 | 20 | 25 | 30 | High |

* **ID :** Indique l’ID de la mesure, attribué à la mesure par la carte d’acquisition.
* **Type :** Indique le type de mesure (CO², Temperature, Humidity)
* **Data :** Indique la donnée convertie, « - » si ID non configuré
* **Critical Min:** Seuil minimum en-dessous duquel le status d’alarme est critique
* **Warning Min:** Seuil minimum en-dessous duquel le status d’alarme demande attention
* **Critical Max:** Seuil maximum au-dessus duquel le status d’alarme est critique
* **Warning Max:** Seuil maximum au-dessus duquel le status d’alarme demande attention
* **Status :** Indique l’état de l’alarme utilisateur (Too Low – Low – Ok – High – Too High)

## Affichage de mesures en graphique

L’interface doit comprendre un affichage graphique des mesures configurées. Le graphique doit permettre :

* La sélection d’une mesure configurée à représenter
* Pour cette mesure, le choix de la base de temps :
  + en secondes : représentation des 60 derniers échantillons des 60 dernières secondes
  + en minutes : représentation des 60 derniers échantillons minutes (1 mesure minute = moyenne des 60 derniers échantillons seconde)
  + en heures : représentation des 24 derniers échantillons heures (1 mesure heure = moyenne des 60 derniers échantillons minute)
* La représentation des limites haute et basse « warning » et « critical » pour la mesure sélectionnée.

## Gestion de comptes utilisateurs

Le programme doit contenir un Dataset « UserAccount » contenant deux DataTables: « AccessTable » et « UserTable ». Ces tables contiennent les propriétés suivantes:

|  |  |  |
| --- | --- | --- |
| **Propriété** | **Type** | **DataRelation** |
| UserTable | | |
| ID | Int32 | Clé primaire – non nulle |
| UserName | String(20) | Index UNIQUE – non nulle |
| UserPassword | String(20) |  |
| Access\_ID | Int32 | Clé étrangère vers AccessTable |
| AccessTable | | |
| ID | Int32 | Clé primaire – non nulle |
| Name | String20 | non nulle |
| AllowCreateID | Boolean | non nulle |
| AllowDestroyID | Boolean | non nulle |
| AllowConfigAlarm | Boolean | non nulle |
| UserCreation | Boolean | non nulle |

Les données de la table “AccessTable” sont:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Name** | **AllowCreateID** | **AllowDestroyID** | **AllowConfigAlarm** | **UserCreation** |
| 0 | AdminRights | true | true | true | true |
| 1 | MasterRights | true | true | true | false |
| 2 | NoRights | false | false | false | false |

L’interface doit comprendre une page de login utilisateur ainsi qu’une page de création utilisateur, réservée à l’administrateur (AccessID 0). Ces 2 pages doivent être liées à la datatable « UserTable ».

L’interface doit gérer l’activation/désactivation de fonctionnalités liées aux comptes utilisateurs suivant les indications décrites dans le tableau « AccessTable » ci-dessus.

# La sauvegarde en fichiers

L’interface doit prévoir

* la sauvegarde des paramètres configurables dans un fichier de type CSV.
* Le chargement des paramètres configurables depuis un fichier de type CSV.

# La sauvegarde en base de données

Les données propres aux comptes utilisateurs doivent être sauvegardées en base de données suivant la description faite dans l’image ci-dessous .

Les mesures CAPTEURS doivent être sauvegardées en base de données toutes les 5 minutes, suivant la description faite dans l’image ci-dessous .

La base de données est locale et de type ACCESS.

