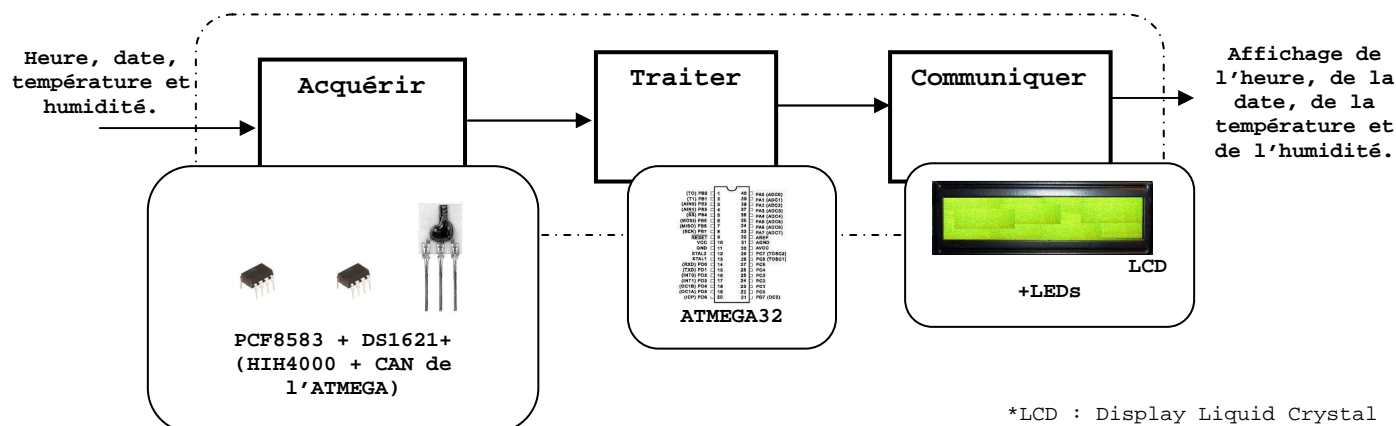


A) Mise en situation

On souhaite **afficher l'heure et la date en plus de la température et de l'humidité**. L'heure et la date seront obtenues à partir d'une horloge temps réel. Ce composant servira également de **référence de temps** pour déclencher les mesures avec une **périodicité** adaptée aux capteurs utilisés. Pour cela, on propose de mettre en œuvre une structure correspondant à la chaîne d'information donnée ci-dessous.



La fonction « Acquérir » est réalisée avec une **horloge temps réel PCF8583**, un capteur numérique de température **DS1621** et un capteur d'humidité **HIH4000** associé au convertisseur analogique numérique du microcontrôleur. La fonction « Traiter » est assurée par un programme implanté dans le microcontrôleur (ATMEGA32). La fonction « Communiquer » est remplie par un afficheur LCD et des LEDs.

L'ensemble des structures matérielles étant réunies sur la carte « SISI » (voir annexe 1), votre travail va se limiter à la **réalisation du logiciel** à implanter dans le microcontrôleur.

Pour cela, vous allez établir l'algorithme d'un programme permettant de **déclencher la lecture des grandeurs physiques** (température et humidité) avec une périodicité **différente** puis vous traduirez cet algorithme en langage C avant de l'intégrer dans le fichier `SERRE0.c` **préparé** à cet effet.

La suite de ce document décrit le travail à réaliser étape par étape. A la fin de cette activité, vous aurez écrit la **première version** du programme « SERRE ».

B) Travail demandé

B1) Ecriture de l'algorithme de mesure des grandeurs physiques

La mesure de la température extérieure à la serre avec un capteur DS1621 et la mesure de l'humidité dans la serre avec un capteur HIH4000 ont été réalisées précédemment (voir FG1 et FG3)

On souhaite maintenant effectuer ces mesures avec une **périodicité différente**.

Exemple

On mesure la température extérieure à la serre toute les **15 mn** et l'humidité intérieure toutes les **5 mn**.

Ceci peut être obtenu en utilisant des compteurs chargés de régler la période de chacune des mesures. En incrémentant ces compteurs toutes les secondes, on dispose d'une valeur représentative du temps écoulé entre chaque mesure.

La mesure du temps est obtenue à partir d'une **horloge temps réel**.

L'algorithme à réaliser peut être décrit par les opérations suivantes :

- **1 Mesurer** le temps (en secondes)
- **2 Incrémenter** les compteurs (associés aux mesures) si **1s** s'est écoulée depuis la dernière mesure.
- **3 Tester** les compteurs et effectuer une mesure si la période de mesure est écoulée.

Pour simplifier l'écriture de cet algorithme, vous disposez des **fonctions suivantes** :

- **LireTemperatureExtérieure()** : renvoie la valeur de la température en °C
- **LireHumidite()** : renvoie la valeur de l'humidité en %

Vous utiliserez les **variables** décrites dans le tableau ci-dessous :

Nom	Type	Description
CptMesTempExt	Entier non signé sur 16 bit	Compteur associé à la mesure de la température extérieure à la serre (contient le nombre de secondes écoulées depuis la dernière mesure)
CptMesHumidite	Entier non signé sur 16 bit	Compteur associé à la mesure de l'humidité intérieure à la serre. (contient le nombre de secondes écoulées depuis la dernière mesure)
Temp_n	Octet	Contient la température mesurée
RH_n	Octet	Contient l'humidité mesurée

Q1) Ecrivez l'algorithme ou dessinez l'algorithme de mesure des grandeurs physiques. (Uniquement la troisième opération dans la liste ci-dessus)

[illegible]

Appel prof

B2) Recherche documentaire sur le composant « Horloge temps réel » PCF8583

B21) Généralités

Q2) Quel est le rôle d'une horloge temps réel ?

B22) Etude de la documentation du composant PCF8583

Q3) Décrivez en une phrase la fonction du circuit PCF8583

Q4) Quelles sont les informations (en rapport avec un calendrier et une horloge) contenues et automatiquement remises à jour dans le circuit PCF8583. (voir le « Block Diagram » du datasheet PCF8583)

Q5) En dehors de ses fonctions d'horloge et de calendrier, que peut-on faire avec ce circuit ?

Q6) Combien de composant PCF8583 peut-on placer sur un bus I2C ?

On souhaite accéder à un composant PCF8583 dont la broche A0 est reliée au 0V.

Q7) Déterminez l'adresse <SLA> de ce composant et les octets <SLA+W> et <SLA+R>. Donnez vos résultats en binaire et en hexadécimal.

A ce stade du projet, on utilise quatre circuits I2C : une **HTR** PCF8583, un **capteur de température** DS1621 et **deux ports d'entrées sorties** PCF8574. Pour les évolutions futures de ce projet, il est utile de disposer d'un plan mémoire (mapping memory). En effet, deux circuits ne pouvant pas disposer de la même adresse, il est nécessaire d'avoir une vision globale de celles qui sont utilisées pour procéder à d'éventuels changements d'adresses.

Le plan mémoire ci-dessous sera réutilisé lorsque vous remplacerez le capteur **analogique HIH4000** par un capteur **numérique SHT11** délivrant également la température.

Q8) Complétez le plan mémoire ci-dessous (h = hexadécimal) avec l'adresse de l'HTR.

<SLA>	Esclave I2C
48h	Capteur de température DS1621
21h	BP sur carte IHM
20h	Led sur carte IHM

B23) Utilisation de la bibliothèque PCF8583 du logiciel CVAVR

Ouvrez la rubrique d'aide en ligne représentée

par l'icône . Clic

Sélectionnez :

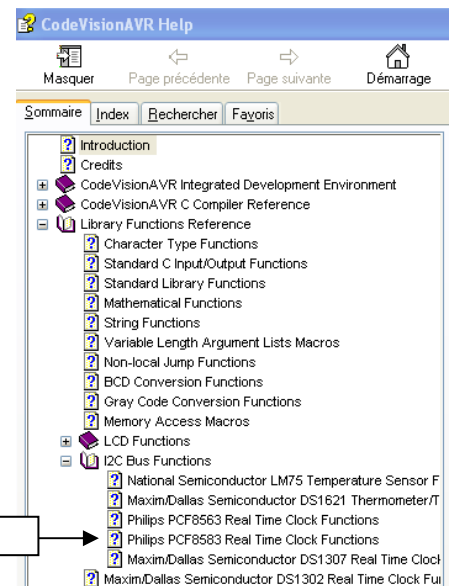
"**Library Functions** Reference"


puis

"**I2C Bus Functions**"

puis

"**Philips PCF8583 Real Time Clock Functions**"



 En langage C, les fonctions comprennent trois parties :

<Le type de la variable renvoyée> <Le nom de la fonction> <la liste des paramètres>
par la fonction passés à la fonction

Void i2c_init ((void))

Q9) Quelle est le nom de la fonction permettant de lire l'heure sur le PCF8583 ? Quel est l'utilité du paramètre **unsigned char chip** ?

Q10) Quelle valeur doit-on attribuer au paramètre **unsigned char chip** pour communiquer avec le circuit PCF8583 monté sur la carte SISI (voir le schéma)? Quelle simplification est apportée par les fonctions de la bibliothèque **PCF8583** lors de l'adressage du circuit ?

B3) Ecriture du programme Serre0

Malgré la complexité des structures à mettre en œuvre, le programme à réaliser reste relativement facile à écrire. Ceci est dû à la « richesse » des **bibliothèques de fonctions** fournies avec le cross-compileur CodeVisionAVR.

Exemple : la lecture de l'heure fournie par l'horloge temps réel s'effectue avec la fonction **rtc_get_time**:

```
rtc_get_time(adresse_HTR,&heures,&minutes,&secondes_n,&dizieme);
```

adresse_HTR correspond à l'adresse du composant (état logique de la broche A0). **Après la lecture, &heures, &minutes, &secondes_n, &dizieme** contiennent respectivement l'heure, les minutes, les secondes et les dixièmes de seconde.

On accède à cette fonction si une **référence** au fichier *pcf8583.h* est placée dans le fichier source *.c* du programme.

Une référence à une bibliothèque se fait à l'aide de la **directive de compilation**

```
#include <nom bibliothèque>.
```

Exemple : `#include < pcf8583.h >`

La lecture de la date fournie par l'horloge temps réel s'effectue avec la fonction

```
rtc_get_date(adresse_HTR,&jour,&mois,&annee);
```

L'écriture sur le LCD nécessite l'utilisation de la fonction `sprintf()`. Cette fonction est située dans la librairie `stdio` accessible par une référence au fichier `stdio.h`

La **partie du programme** à compléter correspond à **votre algorithme**, elle doit uniquement exécuter l'action ci-dessous:

3- Tester les compteurs et effectuer une mesure si la période de mesure est écoulée

Copiez le répertoire du projet « Serre0 » sur le bureau du PC.

Ouvrez le fichier projet « **SERRE0.prj** » avec **CodeVisionAVR**



Q11) Traduisez votre algorithme en langage C et **complétez** la partie du fichier source « **SERRE0.c** » identifiée ci-dessous.

```
//-----  
// 3 Test des compteurs et mesure si la période de mesure s'est écoulé !!!!!!! A compléter !!!!!!!  
//-----  
// Acquisition de la température extérieure à la serre  
  
    // à compléter  
    {  
        // à compléter  
        // à compléter  
  
        // Affichage de l'acquisition de la température extérieure sur la Led D1  
        Ecrire_Port_I2C_Soft(Add_LED_I2C,VisuLectureTempExt);  
        // Visualisation de la prise de mesure  
        delay_ms(50);  
        Ecrire_Port_I2C_Soft(Add_LED_I2C,LedsEteintes);  
    }  
  
// Acquisition de l'humidité intérieure à la serre  
  
    // à compléter  
    {  
        // à compléter  
        // à compléter  
        // à compléter  
  
        // Affichage de l'acquisition de l'humidité intérieure sur la led D3  
        Ecrire_Port_I2C_Soft(Add_LED_I2C,VisuLectureHumid);  
        // Visualisation de la prise de mesure  
        delay_ms(50);  
        Ecrire_Port_I2C_Soft(Add_LED_I2C,LedsEteintes);  
    }  
    } // fin de la mesure des grandeurs physiques  
//-----
```

Appel prof

Pour faire vérifier le fonctionnement du programme

C) Programmation du composant

Compilez le projet par l'icône "Build the Project"



Téléchargez le programme dans la carte SISI (Bouton « Program »)



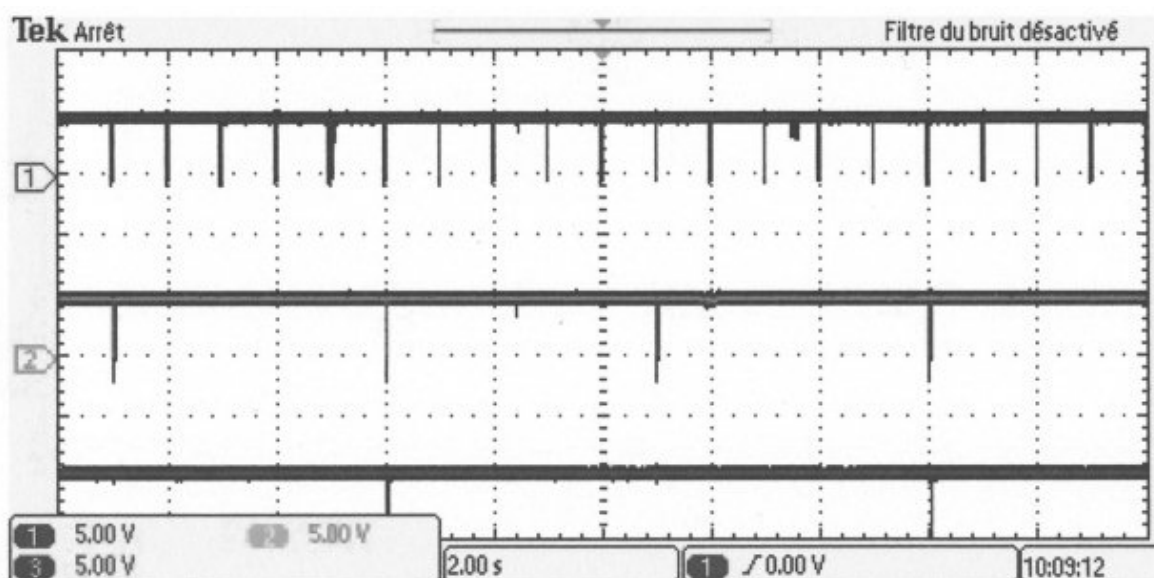
Voir l'annexe 2 pour l'utilisation des menus.



D) Test du programme

Q12) Proposez une méthode permettant visualiser la périodicité des mesures.

Les chronogrammes ci-dessous ont été mesurés lors du test du programme sur la carte SISI.



- Voie 1 : Commande de la Led D0 (Mesure du temps)
- Voie 2 : Commande de la Led D2 (Mesure de température extérieure à la serre)
- Voie 3 : Commande de la Led D3 (Mesure de l'humidité dans la serre)

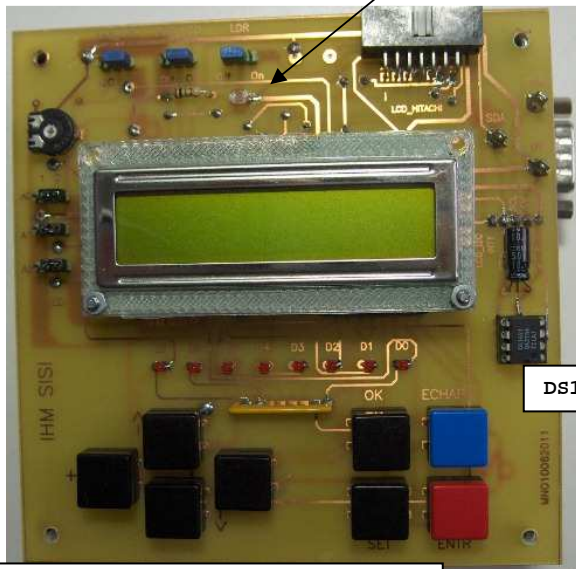
Q13) Déterminez les différentes périodes de mesure et complétez le tableau ci-dessous.

Mesure	Périodicité mesurée
Temps	
Température extérieure	
Humidité intérieure	

Annexe 1 : Module SISI

Carte IHM

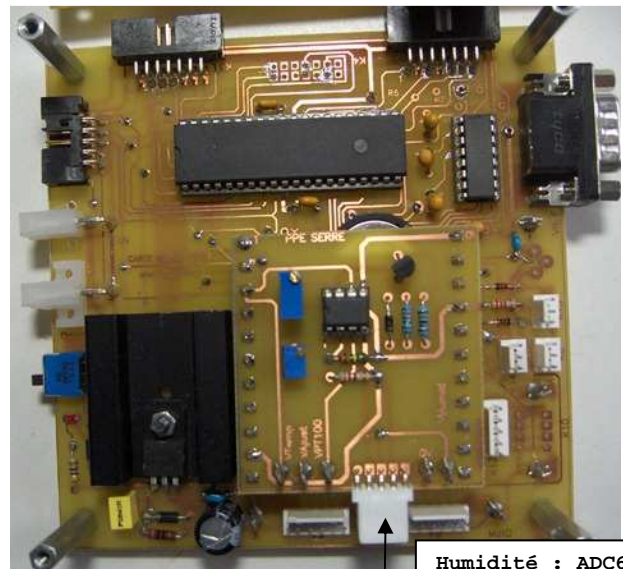
LDR



DS1621

<sla>DS1621 : 48h
<sla>Led : 20h à 27h (20h)
<sla>BP: 20h à 27h (21h)

Carte micro



Humidité + Temp. Int

Humidité : ADC6
Temp. Int : ADC5
Luminosité : ADC0

Annexe 2 : Graphe d'états-transitions des menus du programme « SERRE0 »

