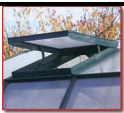




Fiche guide M1	TS SI		P.P.E. Mini serre	
Mesure	2h			
 Lycée Polyvalent PIERRE EMILE MARTIN	Mesurer et afficher une température			

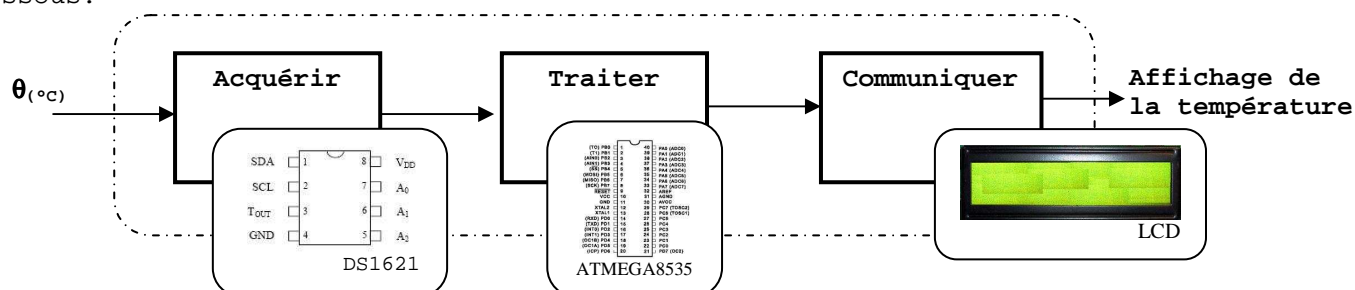
Nom :	Classe :	Groupe :
-------	----------	----------

Objectif : A la fin de ce TP, vous serez capable de **mesurer la température** dans la mini serre.

<p>Matériels Carte ATMELESSI V1. Alimentation 10V.</p> <p>Logiciels CodeVisionAvr.</p> <p>Documentation Schéma de la carte ATMELESSI. Cours et TD sur le bus I²C. Documentation technique des composants DS1621 et afficheur LCD à processeur Hitachi</p> <p>Le présent document et la documentation technique des composants sont <u>téléchargeables</u> sur le site <u>WebGE</u> à l'adresse http://p.mariano.free.fr/ (rubrique PPE)</p>

A) Présentation

On souhaite **afficher la température ambiante** sur un LCD*. Pour cela, on propose de mettre en œuvre une structure correspondant à la chaîne d'information donnée ci-dessous.





La fonction « Acquérir » est réalisée par un capteur numérique de température (DS1621). La fonction « Traiter » est assurée par un programme implanté dans un microcontrôleur (ATMEGA8535). La fonction « communiquer » est remplie par un afficheur LCD (à processeur Hitachi).

L'ensemble des structures matérielles étant réunies sur la carte ATMELESSI, votre travail va se limiter à la **réalisation du logiciel** à implanter dans le microcontrôleur de cette carte.

Pour cela, vous allez **créer et configurer un projet** avec le magicien du cross-compileur **CodeVisionAVR** afin d'obtenir la structure de votre programme. Puis vous complèterez cette structure avec les fonctions nécessaires à la mise en œuvre du capteur et de l'afficheur.

La suite de ce document décrit le travail à réaliser étape par étape. A la fin de cette activité, vous serez capable de mesurer la température dans la mini serre.

*LCD : Display Liquid Crystal

 Lycée Polyvalent PIERRE EMILE MARTIN	FGM1	Mesures		PPE Mini Serre	1
--	------	---------	---	----------------	---

Révision le 24/04/2009

B) Travail demandé

Etape 1 : Création et configuration d'un projet

Lancez le logiciel CodeVisionAVR

(1) Création d'un nouveau projet

Dans la barre d'outils : « **File** » puis « **New** » pour obtenir la boîte de dialogue ci-contre.

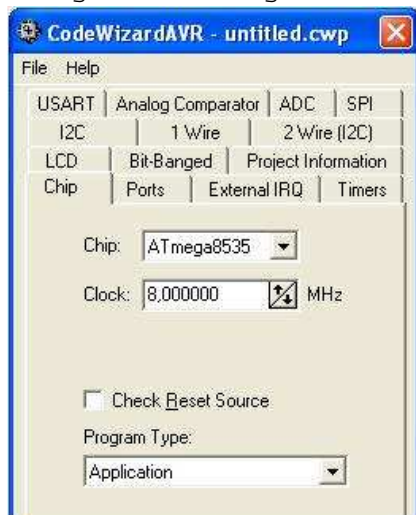
Cochez « **Project** » puis clic sur « **Ok** »



Ici « **Yes** »

(2) Sélection du composant cible

La boîte du « **Magicien** » ci-dessous s'ouvre. Choisissez le « **Chip** » ATmega8535 et réglez le signal d'horloge « **Clock** » à 8Mhz.

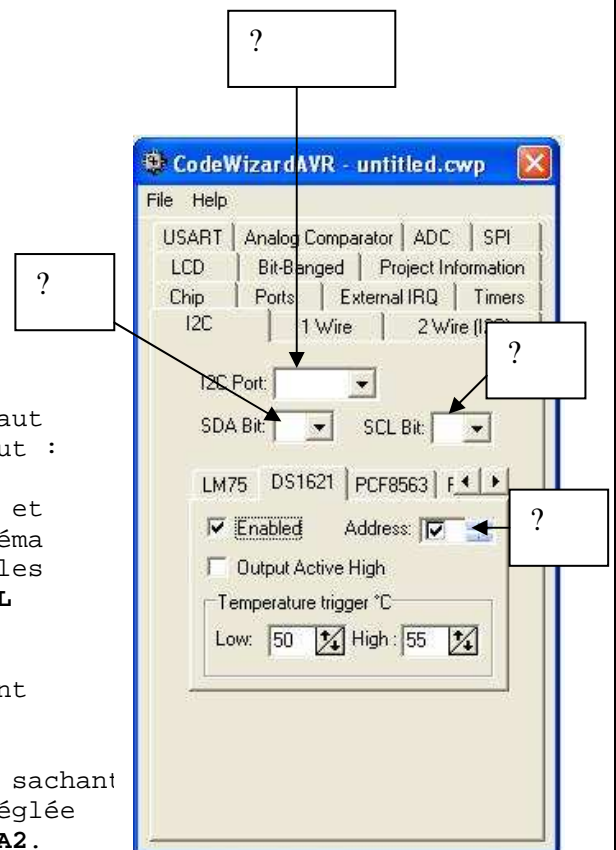


(3) Configuration du bus I2C

Sélectionnez l'onglet « **I2C** ».

Pour que le microcontrôleur ATmega8535 communique avec le capteur DS1621, il faut configurer le bus I2C. Pour cela, il faut :

- **Identifier** la position des lignes SCL et SDA sur le microcontrôleur (voir le schéma de la carte **ATMELSSI V1**) et configurer les champs « **I2C Port** », « **SDA Bit** » et « **SCL Bit** » dans la boîte de dialogue.
- **Sélectionnez** l'onglet DS1621 en cochant « **Enable** ».
- **Régler** le champ « **Adresse** » du DS1621 sachant que l'adresse du composant DS1621 est réglée avec les broches identifiées **A0**, **A1** et **A2**.



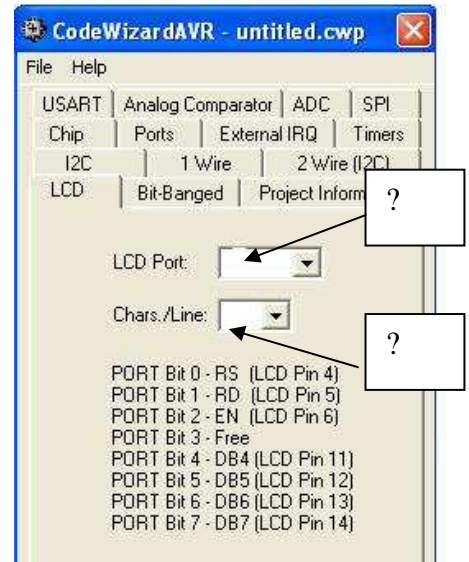
(4) Choix de l'affichage

Sélectionnez l'onglet « LCD ».

Pour obtenir l'organisation ci-contre, il est nécessaire de remplir le champ « LCD Port ».

En étudiant le schéma de la carte « **ATMELSSI V1** », et la documentation de l'afficheur LCD, **déterminez** sur quel port est connecté l'afficheur et son nombre de caractères par ligne.

Configurez les champs « LCD Port » et « Chars./Line » de la boîte de dialogue « LCD ».



(5) Enregistrement du projet



Sélectionnez « Program Preview ».

Si le projet est correctement configuré, le début du fichier source du programme doit correspondre au texte ci-dessous.

```
#include <mega8535.h>

// I2C Bus functions
#asm
    .equ __i2c_port=0x12
    .equ __sda_bit=3
    .equ __scl_bit=7
#endasm
#include <i2c.h>

// DS1621 Thermometer/Thermostat functions
#include <ds1621.h>

// Alphanumeric LCD Module functions
#asm
    .equ __lcd_port=0x15
#endasm
#include <lcd.h>
```

Fermez la fenêtre.

Sélectionnez

→ File

→ Generate, save and Exit

Donnez le nom **Temp** à votre projet (3 fois) pour créer les trois fichiers à la base du projet. (.c, .prj, .cwp)

ATTENTION : Le Magicien ne peut plus être utilisé pour modifier votre projet. Voir le prof pour d'éventuelles corrections.

Etape 2 : Mesure et affichage de la température

Dans ce paragraphe, vous allez compléter la **partie déclarative** ...

```
void main(void)
{
// Declare your local variables here
```

... et la partie **exécutive** du programme.

```
while (1)
{
// Place your code here

};
```

On rappelle que la **partie déclarative** d'un programme est la zone dans laquelle sont **créées les variables** alors que la **partie exécutive** est la zone de **traitement** de ces variables.

Malgré la complexité des structures à mettre en œuvre, le programme à réaliser reste relativement simple. Ceci est dû à la « richesse » des **bibliothèques de fonctions** fournies avec le cross-compileur CodeVisionAVR.
Par exemple, la lecture de l'information fournie par le capteur DS1621 s'effectue avec la fonction :

ds1621_temperature_10(?)

? correspond à l'adresse du composant. Cette fonction est située dans la bibliothèque I2C. Elle est accessible lorsqu'une **référence** au fichier <I2C.h> **est présente** dans le programme.

L'écriture sur le LCD nécessite la fonction **sprintf()**. Cette fonction est située dans la bibliothèque stdio accessible par une référence au fichier <stdio.h>

(1) Déclaration des bibliothèques de fonctions utilisées dans le programme

La référence à la bibliothèque I2C a été automatiquement placée par le magicien avec la directive `#include <i2c.h>` (voir l'en-tête du programme).

Par contre, vous **devez rajouter** une référence à la bibliothèque stdio à la suite de `#include <mega8535.h>` dans le fichier source. Vous **rajouterez** également les références aux bibliothèques *math* (nécessaire pour effectuer un calcul de valeur absolue) et *delay* (nécessaire pour réaliser une temporisation).

(2) Partie exécutive du programme à réaliser

Le programme à réaliser se limite à la répétition de deux actions :

Lire (température)
Ecrire (température)

o **Acquisition de la température**

L'acquisition de la température est faite par l'expression ci-dessous :

temperature = ds1621_temperature_10(?); // L'image de la température mesurée est placée dans la variable température.

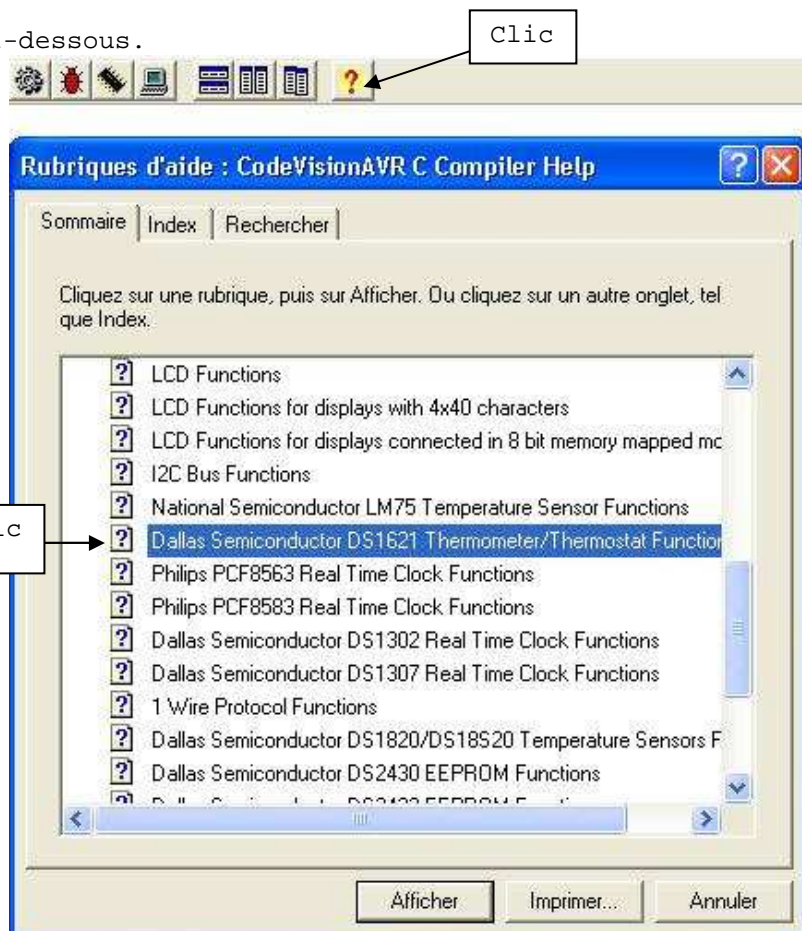
Ouvrez la rubrique d'aide comme ci-dessous.

Clic

Sélectionnez
« CodeVisionAvr C compiler Library
Functions »

Clic sur « Afficher »

Lisez la description de la fonction
repérée par :
`int ds_1621_temperature_10(unsigned
char chip)`



On suppose que la température mesurée est 27°C, quelle sera la valeur contenue dans la variable « temperature » ?

Complétez le fichier source C comme ci-dessous et remplacez le ? par l'adresse du capteur de température.

```
while (1)
{
// ----- Lecture de la température -----

    temperature = ds1621_temperature_10(?);    // à compléter

};
```

o Affichage de la température

La valeur à afficher est placée dans une zone mémoire appelée **buffer** avant d'être envoyée à l'afficheur. Ceci est réalisé par l'expression ci-dessous.

```
sprintf(display_buffer, "T=%-i.%-u%-cC", temperature/10, abs(temperature%10), 0xdf);
```

Expliquez la présence de la division par 10 dans l'expression ci-dessus

Le contenu du buffer est ensuite envoyé à l'afficheur avec la commande ci-dessous

```
lcd_puts(display_buffer);
```

Complétez le fichier source C comme ci-dessous et remplacez ?? pour que la temporisation ait lieu toutes les 1s.

```
while (1)
{
// ----- Lecture de la température -----

temperature = ds1621_temperature_10(?);    // à compléter

// ----- Traitement -----
sprintf(display_buffer,"T=-i.%-u%-cC",temperature/10,abs(temperature%10),0xdf);

// ----- Affichage -----
lcd_clear();
lcd_puts(display_buffer);

delay_ms(??);    // à compléter pour avoir une mesure toutes les secondes

};
```

Quelle est le rôle de la fonction `lcd_clear()` ?

(3) Partie déclarative du programme à réaliser

Le programme ci-dessus utilise deux types de variables :

- **temperature** : un entier de type octet et
- **display_buffer** : un tableau d'octets

Pour être reconnues, ces variables doivent être déclarées avant leur utilisation.

Complétez le fichier source C comme ci-dessous :

```
void main(void)
{
// Declare your local variables here
// -----
//type          nom          Commentaires
// -----
char display_buffer[17];    // Tampon pour la première ligne de l'afficheur
int temperature;           // Image de la température mesurée par le DS1621
```

C) Programmation du composant

Configurez le projet

- > Project
- > Configure
 - > Sélectionnez l'onglet "After Make"
 - > Cochez "Program the Chip"
 - > ok

Programmez le composant



- > Icône "Make the Project"
- > "Program"

Appel prof

Pour faire vérifier le fonctionnement

	FGM1	Mesures		PPE Mini Serre	6
--	------	---------	---	----------------	---