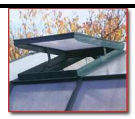
Fiche guide M2	TS SI		P.P.E. Mini serre	
Mesure	2h			
Lycée Polyvalent PIERRE EMILE MARTIN	Détecteur Jour / Nuit			

Nom :	Classe :	Groupe :

Objectif : Etre capable de mesurer et d'afficher deux seuils de luminosité.

Matériels
Carte ATMELSSI V1 + Module capteur LDR + Alimentation 10V.

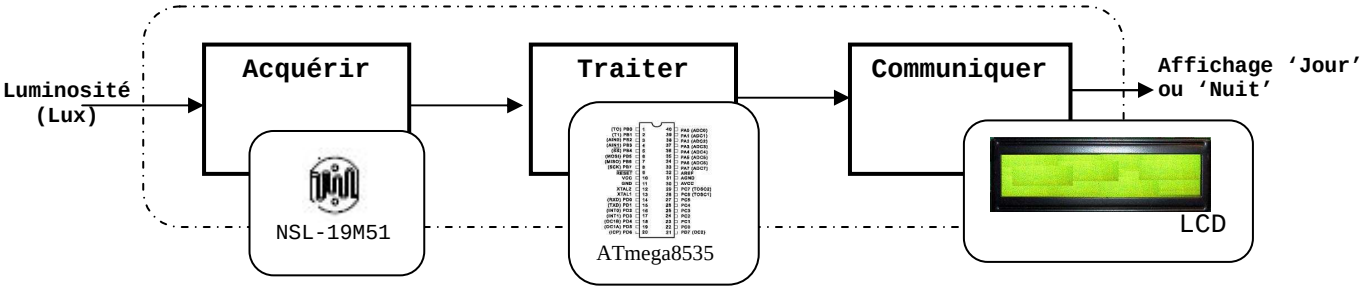
Logiciels
CodeVisionAvr.

Documentation
Schémas de la carte ATMELSSI et de la carte capteurs.
Documentation technique de la LDR NSL-19M51 et de l'afficheur LCD à processeur Hitachi.

Le présent document et la documentation technique des composants sont **téléchargeables** sur le site **WebGE** **à l'adresse** http://p.mariano.free.fr/ **(rubrique PPE)**

A) Présentation

On souhaite **afficher la valeur de la luminosité** ambiante (**deux seuils)** sur un afficheur 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 une LDR* (NSL-19M51) et le convertisseur analogique numérique du microcontrôleur (ATmega8535). La fonction « Traiter » est assurée par un programme implanté dans le microcontrôleur. La fonction « Communiquer » est remplie par un afficheur LCD (processeur Hitachi).

L'ensemble des structures matérielles étant réunies sur la carte « ATMELSSI » et la carte « Capteurs », votre travail va se limiter à la **réalisation du logiciel** à implanter dans le microcontrôleur.

Pour cela, vous allez **créer et configurer un projet** avec le « magicien » du cross-compilateur **CodeVisionAVR**. 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 détecter et d'afficher deux seuils de luminosité.

*LCD : Display Liquid Crystal *LDR : Light Dependant Resistor

B) Travail demandé

Etape 1 : Création et configuration d'un projet

Lancez le logiciel **CodeVisionAVR**

(1) Création d'un nouveau projet

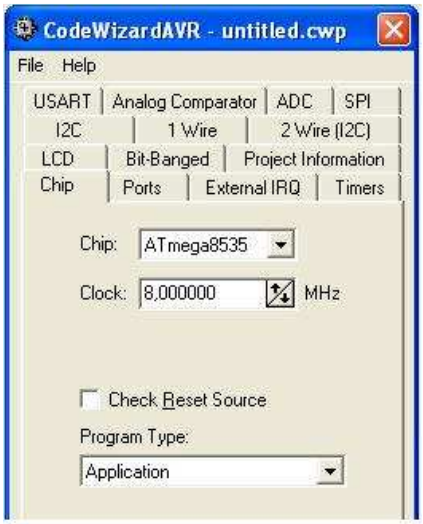
Dans la barre d'outils : **« File »** puis « **New »** pour obtenir la boîte de dialogue ci-contre.

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.

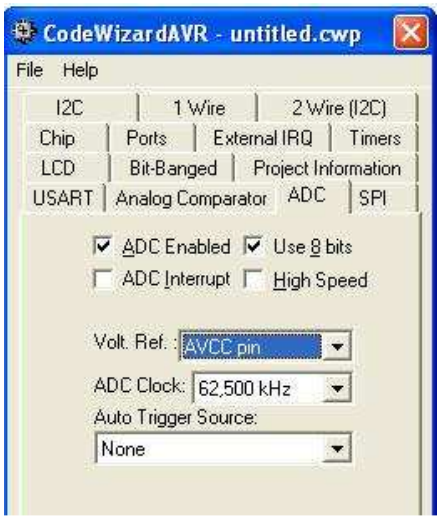


(3) Configuration du convertisseur analogique numérique

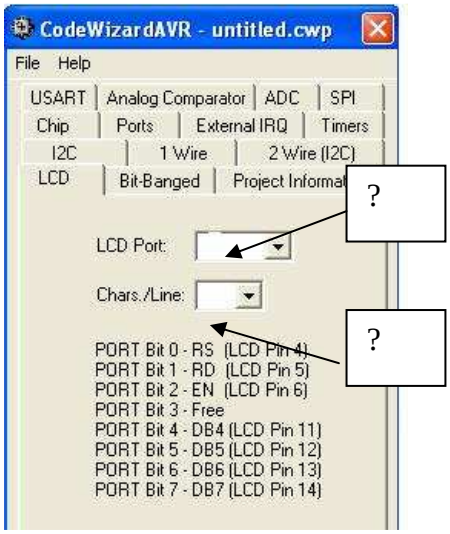
Sélectionnez l'onglet « **ADC** ».

Il est nécessaire de cocher la case « **ADC Enabled** » et de régler l'horloge « **ADC Clock** » à 62,5kHz pour utiliser le convertisseur analogique numérique du microcontrôleur.

Précisez également que le résultat de la conversion doit être fourni sur **8 BIT**.



(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 boite 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>

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

#define ADC_VREF_TYPE 0x60
// Read the 8 most significant bits
// of the AD conversion result
unsigned char read_adc(unsigned char adc_input)
{
ADMUX=adc_input|ADC_VREF_TYPE;
// Start the AD conversion
ADCSRA|=0x40;
// Wait for the AD conversion to complete
while ((ADCSRA & 0x10)==0);
ADCSRA|=0x10;
return ADCH;
}
```

Fermez la fenêtre.

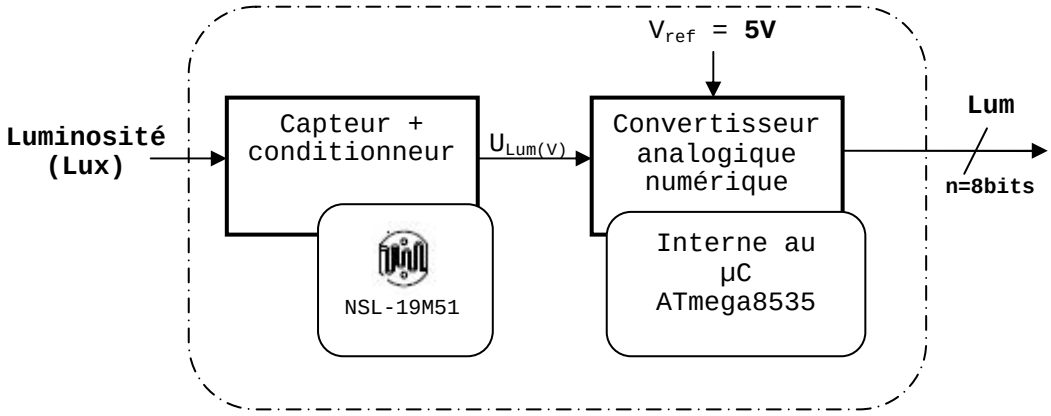
Sélectionnez
→ File
→ **Generate, save and Exit**

Donnez le nom **JourNuit** 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 : Etude de la fonction « Acquérir »

La fonction « Acquérir » peut être représentée par le schéma ci-dessous :



Remarque : Lum est une variable entière non signée de type octet.

- **Détermination des valeurs de Lum représentatives du 'Jour' et de la 'Nuit'**

  Le programme à réaliser doit afficher 'Jour' ou 'Nuit' en fonction de la valeur de la variable Lum. Celle-ci dépend de la luminosité ambiante.

  La luminosité est mesurée par une **photorésistance** (**LDR**) type NSL-19M51.

  Expliquez le fonctionnement de ce type de capteur. (Recherches nécessaires)
  Remarques : Expliquez notamment comment évolue sa résistance en fonction de la luminosité.

  _____

  _____

  _____

  _____

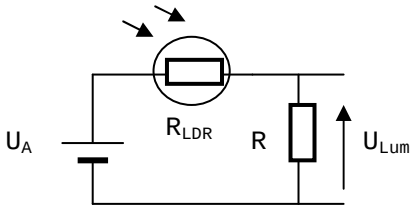
  _____

  _____

  La photorésistance est intégrée à un montage **diviseur de tension** (**conditionneur**)

  Cette structure délivre une différence de potentiel U_{Lum} représentative de la luminosité ambiante. Elle va vous permettre de réaliser un détecteur à seuils utilisé notamment pour autoriser ou non **l'arrosage des plantes**.

  Schéma

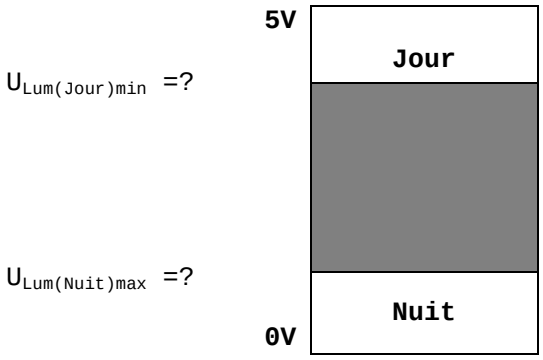


  U_A = 5V R = 1MΩ

Dans la suite de ce paragraphe, vous allez établir les valeurs de U_{Lum} représentatives des informations « Jour » et « Nuit » et représenter le comportement du montage sous la forme d'un gabarit.

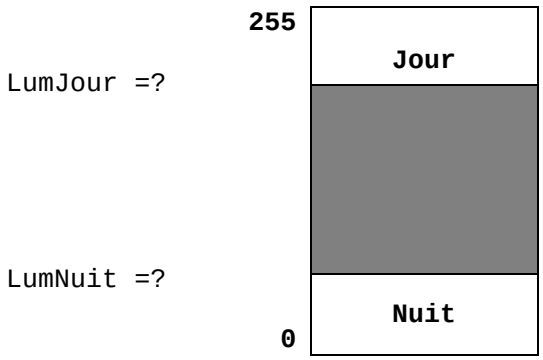
Etude théorique du montage

Exprimez U_{Lum} en fonction de U_A puis **calculez** $U_{Lum(Jour)min}$ et $U_{Lum(Nuit)max}$ en utilisant les paramètres ***Light resistance*** max et ***Dark resistance*** donnés dans la fiche technique de la photorésistance. **Complétez** le gabarit ci-dessous



On appelle **LumJour** la valeur numérique correspondant à $U_{Lum(Jour)min}$ et **LumNuit** celle correspondant à $UL_{um(Nuit)max}$.

Sachant que **Lum = $(2^n/V_{ref}).U_{Lum}$,** calculez LumJour et LumNuit et **complétez** le gabarit ci-dessous.



Vous disposez maintenant de deux valeurs numériques représentatives du jour et de la nuit. Ces valeurs seront à **ajuster expérimentalement** dans votre programme.

Etape 3 : Mesure et affichage de la luminosité

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 rappel 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 du à la « richesse » des **bibliothèques de fonctions** fournies avec le cross-compilateur CodeVisionAVR.

Par exemple, la lecture de l'information fournie par le capteur de luminosité s'effectue avec la fonction :

read_adc(?)

? correspond à l'entrée sélectionnée sur le port A du microcontrôleur.
Cette fonction a été introduite dans le programme par le Magicien (voir page 3 de ce document).

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

On fait référence à une bibliothèque avec la directive **#include**.

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

Vous **devez rajouter** une référence à la bibliothèque stdio à la suite de #include <mega8535.h> dans le fichier source. Vous rajouterez également une référence à la bibliothèque delay (nécessaire pour réaliser une temporisation).

(2) Partie exécutive du programme à réaliser

Le programme à réaliser peut être résumé par les actions ci-dessous:

```
Lire (luminosité)
Traitement (Déterminer s'il fait plutôt jour ou nuit)
Ecrire ('Jour' ou 'Nuit')
```

- **Acquisition de la luminosité**

  La luminosité est lue par l'expression ci-dessous :

  Lum = read_adc(?);

  ? correspond au numéro de la broche du Port A sur laquelle est connectée le capteur. Voir le schéma de la carte « Capteurs ».

Complétez le fichier source C comme ci-dessous et remplacez le ? par le numéro de la broche du Port A connectée au capteur de luminosité.

Remplacez ?? pour avoir une temporisation de 10s.

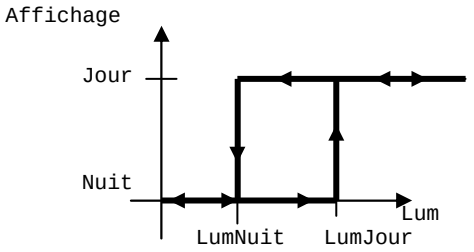
```
delay_ms(??);// Attente de 10s avant la première mesure

while (1)
      {
// -------------------- Lecture de la luminosité -------------------

        Lum = read_adc(?);                // A compléter
      };
```

- **Traitement et affichage de la luminosité**

Le programme à réaliser doit afficher « Jour » ou « Nuit » sur le LCD en fonction de la luminosité ambiante. Ce programme peut être représenté par le cycle ci-dessous.



Etablissez l'algorigramme correspondant au cycle ci-dessus.
Indications : Utiliser une des structures alternatives (si alors … sinon ou selon …). Vous disposez également d'une fonction Affiche(« Texte »).

Algorigramme

Début

Dans le langage C du cross compilateur CodeVisionAVR, la fonction affiche est réalisée avec :

```
{
      lcd_clear() ;
      sprintf(display_buffer,"Jour");
      lcd_puts(display_buffer);
}
```

ou

```
{
      lcd_clear() ;
      sprintf(display_buffer,"Nuit");
      lcd_puts(display_buffer);
}
```

Traduisez votre algorithme en C et complétez le fichier source C ci-dessous.

```
lcd_clear();
sprintf(display_buffer,"Jour ou Nuit");
lcd_puts(display_buffer);

delay_ms(??);// Attente de 10s avant la première mesure

while (1)
      {
// ------------------------ Lecture de la luminosité ---------------------------

            Lum = read_adc(?);              // A compléter

// ------------------------ Traitement et affichage----------------------------

                                 // A compléter









            delay_ms(??);// Attente de 10s avant la mesure suivante
      };
```

Consultez la documentation de la **photorésistance** (**LDR**) type NSL-19M51NSL et **expliquez** pourquoi la temporisation a été choisie égale à 10s.

_____

_____

_____

_____

(3) Partie déclarative du programme à réaliser

Le programme ci-dessus utilise deux types de variable :
- **Lum** : un entier de type octet,
- **display_buffer** : un tableau de caractères

Pour être reconnue, 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
// ------------------------------------------------------------------------
unsigned char Lum = 0;                        // Image de la luminosité [0, 255]
unsigned char display_buffer[17];             // Tampon ligne 0 de l'afficheur
```

Le programme ci-dessus utilise deux constantes :
- **LumJour et LumNuit**

Pour être reconnue, ces constantes doivent être **déclarées** avant leur utilisation.

Complétez le fichier source C comme ci-dessous :

```
#include <mega8535.h>
                    // bibliothèques à rajouter


#define LumJour      ? // A compléter
#define LumNuit      ? // A compléter

// Alphanumeric LCD Module functions
```

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"

Ajustez les valeurs des constantes **LumJour et LumNuit** en fonction de l'éclairement ambiant.

Pour faire vérifier le fonctionnement

Annexe 1 : Câblage de la carte capteurs