

Fiche élève : Luminosité	TS SI		P.P.E Mini serre	
	Détecteur jour / nuit			

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

Objectif : Afficher « Jour » ou « Nuit » sur un LCD selon la luminosité ambiante.

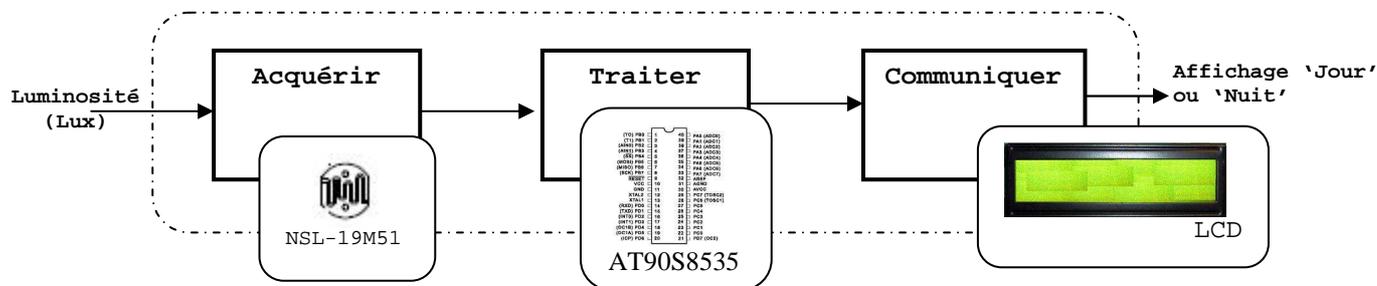
Matériels
Carte ATMELSSI V1 + Module capteur + Alimentation 10V.

Logiciels
CodeVisionAvr.

Sur le site [WebGE](http://p.mariano.free.fr/) à l'adresse <http://p.mariano.free.fr/> (rubrique PPE)
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

A) Présentation

On souhaite **afficher l'humidité** ambiante sur un LCD. Pour cela, on se 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 (AT90S8535). 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-compileur **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 deux seuils de luminosité dans la mini serre.

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

Fiche Détecteur Jour Nuit	Détection du jour et de la nuit avec une LDR et affichage sur un LCD	1
------------------------------	--	---

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** » s'ouvre comme ci-dessous. Choisissez le « **Chip** » AT90S8535 et réglez le signal d'horloge « **Clock** » à 4Mhz.



(3) Configuration du convertisseur analogique numérique

Sélectionnez l'onglet « **ADC** ».

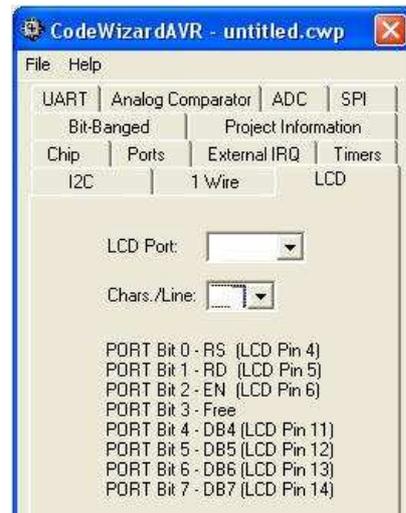
Il est nécessaire de cocher la case « **ADC Enabled** » et de régler l'horloge « **ADC Clock** » à 31,25kHz pour utiliser le convertisseur analogique numérique du microcontrôleur.



(4) Choix de l'affichage

Sélectionnez l'onglet « LCD ».

Après avoir étudié le schéma de la carte « **ATMELSSI V1** », **déterminez** sur quel port est connecté l'afficheur LCD et le nombre de caractères par ligne que comporte cet afficheur. 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 <90s8535.h>

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

// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input;
    // Start the AD conversion
    ADCSR|=0x40;
    // Wait for the AD conversion to complete
    while ((ADCSR & 0x10)==0);
    ADCSR|=0x10;
    return ADCW;
}
```

Fermez la fenêtre.

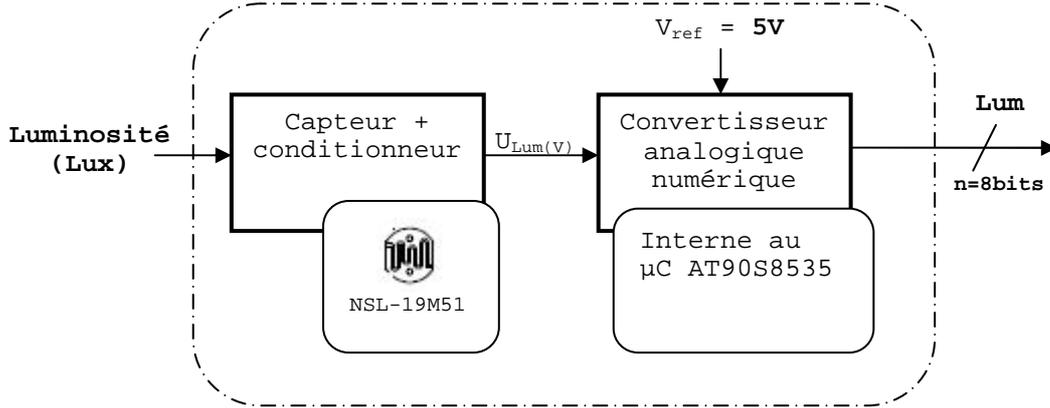
Sélectionnez « **Generate, save and Exit** ».

Donnez le nom **JourNuit** à votre projet.

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 :



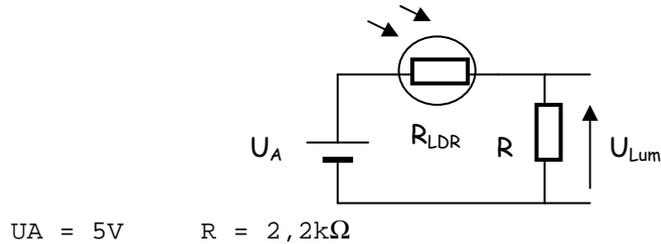
Remarque : Lum est une variable entière non signée de type octet.

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

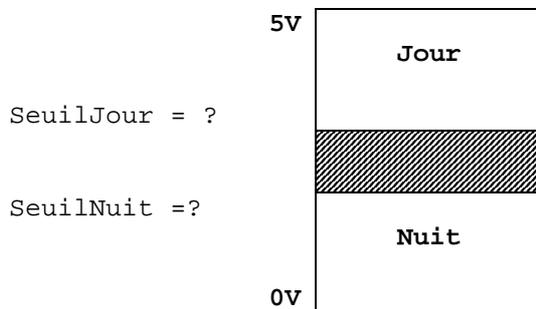
Le programme à réaliser doit afficher 'Jour' ou 'Nuit' en fonction de la valeur de la variable Lum. Celle-ci dépend de la luminosité ambiante.

La luminosité est mesurée par une photorésistance (LDR) intégrée à un montage diviseur de tension (conditionneur). Cette structure délivre une différence de potentiel U_{Lum} représentative de la luminosité ambiante.

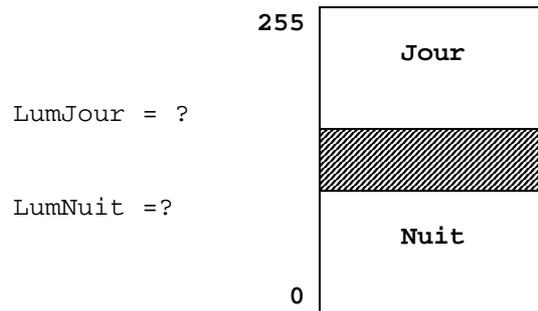
Schéma



Déterminez expérimentalement les tensions correspondant au jour et à la nuit et complétez le gabarit ci-dessous.



On appelle LumJour la valeur numérique correspondant à SeuilJour et LumNuit celle correspondant à SeuilNuit. Sachant que $Lum = (2^n/V_{ref}) \cdot U_{Lum}$, complétez le gabarit ci-dessous.



Etape 3 : Mesure et affichage de la luminosité

Dans cette partie, 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 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 le LCD nécessite la fonction printf(). Cette fonction est située dans la librairie stdio accessible par le fichier stdio.h

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

Vous devez rajouter stdio.h à la suite de #include <90s8535.h> dans le fichier source. Vous ajouterez également la librairie delay.h (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')

o Acquisition de la luminosité

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

```
Lum = read_adc(?)/4;
```

La division par 4 ramène le résultat de la mesure sur 8 bits (le convertisseur analogique numérique du microcontrôleur délivre un résultat sur 10 bits mais seuls les 8 bits de poids fort nous sont utiles)
? 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 5s.

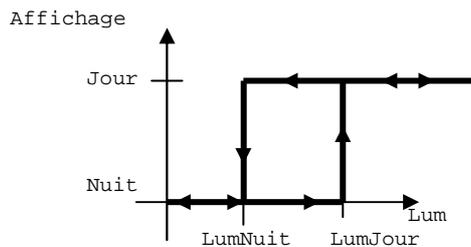
```
delay_ms(??); // Attente de 5s avant la première mesure

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

    Lum = read_adc(?)/4;           // A compléter
};
```

o 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'algorithme ou l'algorithme correspondant au cycle ci-dessus sur la page suivante.

Indications : Utiliser une structure alternative (si alors ... sinon, selon ...). On dispose d'une fonction Affiche(« Texte »).

Algorithme

Dans le langage C du cross compilateur, la fonction affiche se fait 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,"Lum non detectee");  
lcd_puts(display_buffer);  
delay_ms(??); // Attente de 5s avant la première mesure  
  
while (1)  
    {  
// ----- Lecture de la luminosité -----  
  
        Lum = read_adc(?) / 4;           // A compléter  
  
// ----- Traitement et affichage -----  
  
        // A compléter  
  
  
        delay_ms(??); // Attente de 5s avant la mesure suivante  
    };
```

(3) Partie déclarative du programme à réaliser

Le programme ci-dessus utilise deux types de variable :

- Lum : un entier de type octet,
- display_buffer : tableaux 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 <90s8535.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"

Programmez le composant

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