

Fiche guide 5	TS SI		P.P.E Mini serre	 académie d'Orléans-Tours É Éducation nationale enseignement supérieur recherche 
Luminosité	2h			
 Lycée Polyvalent PIERRE EMILE MARTIN	<h2>Détecteur jour / nuit</h2>			

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

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

Matériels
 Carte ATMEL SSI V1 + Carte interface « Capteurs » + Alimentation 10V + Multimètre + Luxmètre.

Logiciels
 CodeVisionAvr V2.

Sur le site WebGE à l'adresse <http://p.mariano.free.fr/> (rubrique PPE)
 Schémas de la carte « ATMEL SSI » et de la carte « Capteurs ».
 Documentation technique de la LDR NSL-19M51 et de l'afficheur LCD à processeur Hitachi

Date : _____

Sommaire

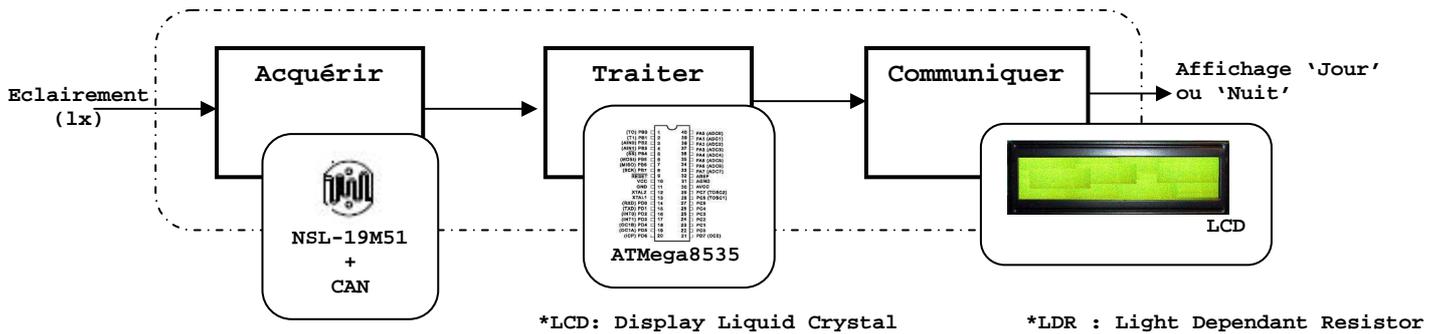
- A) Mise en situation
- B) Travail demandé
 - B1) Recherches documentaires
 - B2) Etude de la fonction « Acquérir »
 - B3) Création et configuration d'un projet avec CVAVR
 - B4) Ecriture du programme
- C) Programmation du composant
- D) Test du programme
- E) Compte rendu

Corrigé le	Commentaires

 Lycée Polyvalent PIERRE EMILE MARTIN	FG5		Luminosité	PPE Mini Serre	1
--	-----	---	------------	----------------	---

A) Mise en situation

On souhaite afficher l'information « Jour » ou « Nuit » selon la luminosité 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 une LDR* (NSL-19M51) et le convertisseur analogique numérique du microcontrôleur (ATMega8535) de la carte SSI. La fonction « Traiter » est assurée par un programme implanté dans ce microcontrôleur. La fonction « Communiquer » est remplie par un afficheur LCD (processeur Hitachi).

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

Après avoir étudié la **fonction acquérir** et déterminé les paramètres nécessaires à votre programme, vous **créerez et configurerez un projet** avec le « magicien » du cross-compileur **CodeVisionAVR**. Vous complèterez ensuite cette structure avec les fonctions nécessaires à la mise en œuvre du capteur et de l'afficheur.

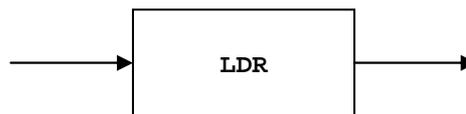
La suite de ce document décrit le travail à réaliser étape par étape. A la fin de cette activité, vous serez capable de détecter et d'afficher deux seuils de luminosité.

B) Travail demandé

B1) Etape 1: Recherches documentaires

Q1) Que signifie LDR ? Expliquez en une phrase le fonctionnement ce composant. Citez une applications des LDR.

Q2) Complétez le schéma ci-dessous par les grandeurs physiques reçue et fournie par la LDR.

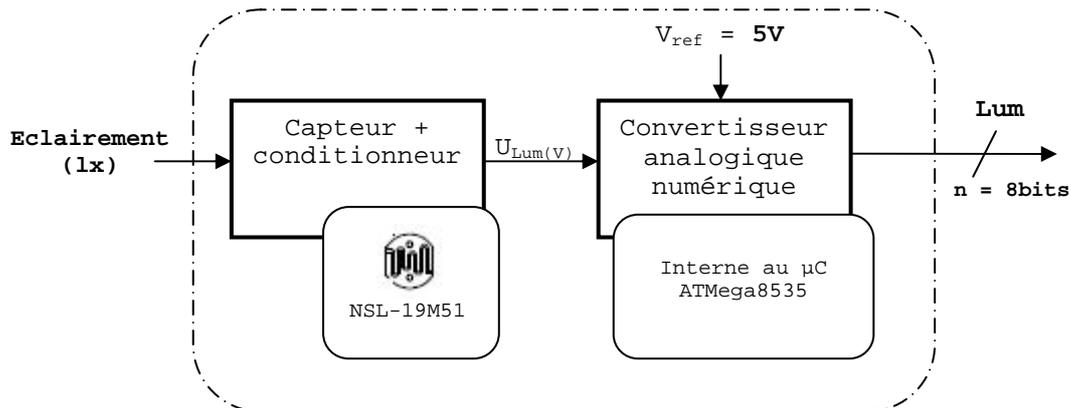


Q3) Vous utiliserez un luxmètre pour mesurer la « lumière » reçue par la LDR. Quelle information délivre cet appareil ? **Donnez** sa définition.



B2) Etape 2: Etude de la fonction « Acquérir »

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



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

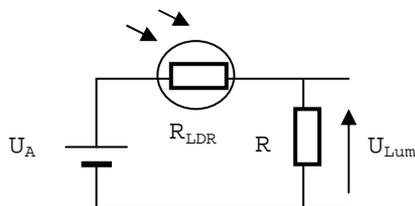
B21) Détermination des valeurs de la variable « Lum » représentatives du "Jour" et de la "Nuit"

Le programme à réaliser doit afficher "**Jour**" ou "**Nuit**" en fonction de la valeur de la variable Lum. Celle-ci dépend de l'éclairement reçu par le capteur.

L'éclairement est mesurée par une photorésistance (LDR) intégrée à un montage diviseur de tension (**conditionneur du capteur**).

Cette structure délivre une différence de potentiel U_{Lum} représentative de la luminosité ambiante.

Schéma



$U_A = 5V$ $R = 1M\Omega$

La photorésistance utilisée est une **NSL-19M51**.

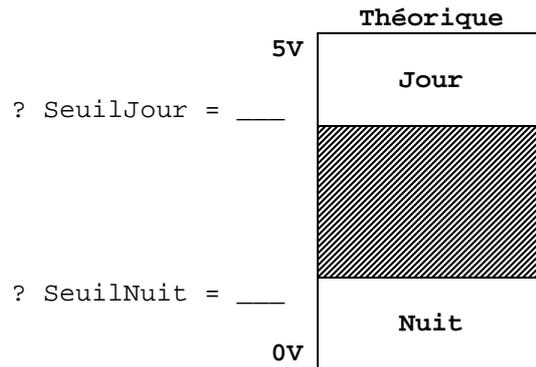
Q4a) Exprimez U_{Lum} en fonction de U_A , R et R_{LDR} .

Q4b) Relevez la valeur maximum de $R_{LDR} = R_L$ (lorsqu'elle reçoit 10lux) sur la documentation de la photorésistance (Annexe 1 de ce document) et la valeur de $R_{LDR} = R_D$ (lorsqu'elle est dans l'obscurité).

10lux => $R_{LDR} = R_{Lmax} =$ _____

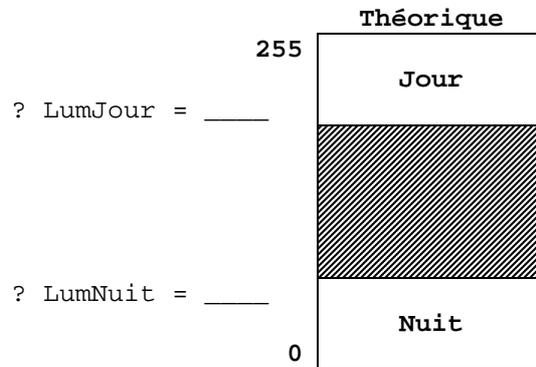
Obscurité => $R_{LDR} = R_{Dmin} =$ _____

Q4c) Calculez U_{Lum} pour ces deux valeurs de R_{LDR} et complétez le gabarit ci-dessous.



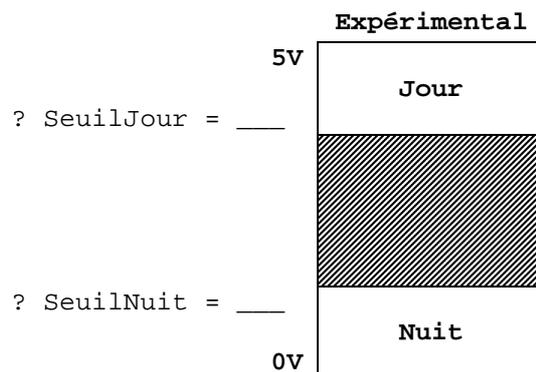
On appelle LumJour la valeur numérique correspondant à SeuilJour et LumNuit celle correspondant à SeuilNuit.

Q4d) Sachant que $Lum = (2^P/V_{ref}) \cdot U_{Lum}$, complétez le gabarit ci-dessous.



Q4e) Déterminez expérimentalement les tensions correspondant au jour et à la nuit et complétez le gabarit ci-dessous.

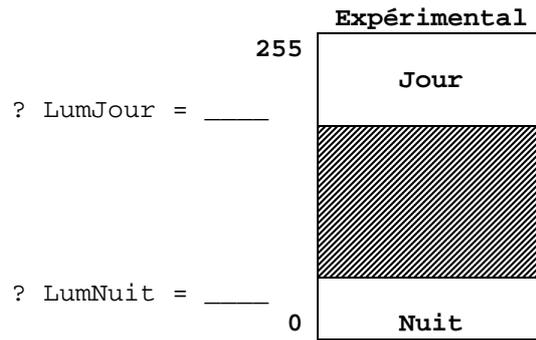
Matériel nécessaire : multimètre



On appelle LumJour la valeur numérique correspondant à SeuilJour et LumNuit celle correspondant à SeuilNuit.



Q4f) Sachant que $Lum = (2^R/V_{ref}) \cdot U_{Lum}$, complétez le gabarit ci-dessous.



Q4g) Effectuez la mesure de luminosité ambiante. Cette mesure nous donne-t-elle une indication sur la valeur de RLDR? Si oui, laquelle ?

Q5) Choisissez les valeurs, à attribuer aux seuils LumJour et LumNuit dans le programme. Justifiez vos choix.

B3) Etape 3: Création et configuration d'un projet avec CVAVR

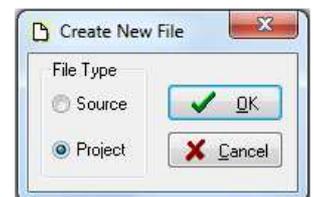
Démarrez le logiciel CodeVisionAVR



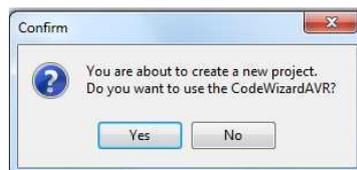
(1) Création d'un nouveau projet

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

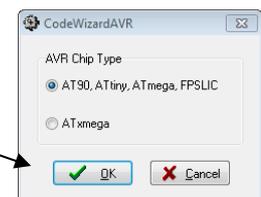
Cochez « **Project** » puis clic sur « **Ok** »



Ici « **Yes** »

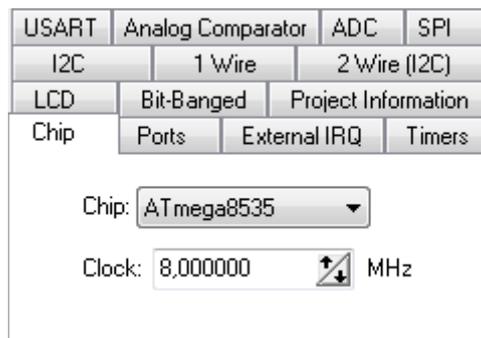


puis « **OK** »



(2) Sélection du composant cible

La boîte du « **Magicien** » (CodeWizardAVR) ci-dessous s'ouvre. **Choisissez** le « **Chip** » ATMEGA8535 et **réglez** le signal d'horloge « **Clock** » à 8Mhz.

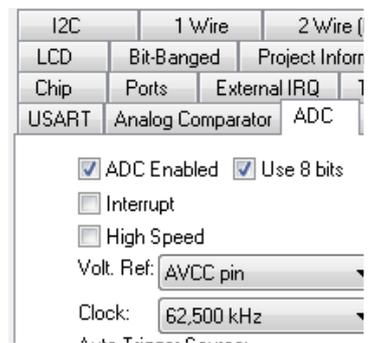


(3) Configuration du convertisseur analogique numérique

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 est fourni sur **8 bits**.

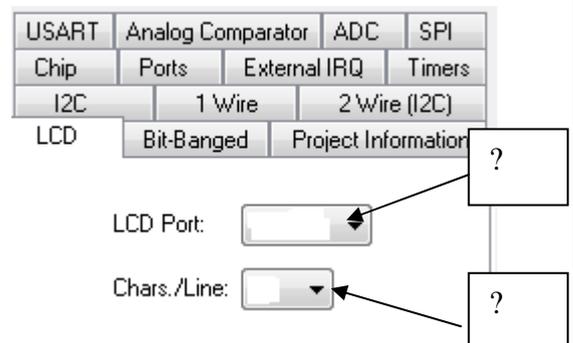


(4) Choix de l'affichage

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 LCD et le nombre de caractères par ligne que comporte cet afficheur.



Q6) 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>

#include <delay.h>

// Alphanumeric LCD Module functions
#asm
    .equ __lcd_port=0x15 ;PORTC
#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 & 0xff);
    // Delay needed for the stabilization of the ADC input voltage
    delay_us(10);
    // Start the AD conversion
    ADCSRA|=0x40;
    // Wait for the AD conversion to complete
    while ((ADCSRA & 0x10)!=0);
    ADCSRA|=0x10;
    return ADCH;
}
```

Sélectionnez

- File
- « **Generate, save and Exit** ».

Donnez le nom **Lum** à votre projet (3 fois) pour créer les trois fichiers de base d'un projet CodeVisionAVR. (le .c, le .prj et le .cwp)



ATTENTION : Après la sauvegarde du projet, le magicien ne peut plus être utilisé pour le modifier. Voir le prof pour d'éventuelles corrections.

B4) Etape 4: Ecriture du programme

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 facile à écrire. Ceci est dû à la « richesse » des **bibliothèques** fournies avec le cross-compileur CodeVisionAVR.

Exemple

La lecture de l'information fournie par la photorésistance 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 bibliothèque stdio. Celle-ci est accessible lorsqu'une **référence** au fichier stdio.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.h>
```

Exemple : #include <stdio.h>

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

Vous devez rajouter <stdio.h> à la suite de #include <ATMega8535.h> dans le fichier source. Vous rajouterez également la librairie <delay.h> (nécessaire pour réaliser une temporisation).

(2) Partie exécutive du programme à réaliser

Le programme à réaliser se compose des actions ci-dessous :

```
Lire (l'éclairement reçu par la photorésistance)
Traitement (Déterminer s'il fait plutôt jour ou nuit)
Ecrire ('Jour' ou 'Nuit')
```

o Acquisition de l'éclairement reçu par la photorésistance

La valeur de l'éclairement est obtenue par l'expression ci-dessous :

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

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

Q7) Complétez le fichier source C comme ci-dessous et remplacez le ? par le numéro de la broche du Port A connectée à la photorésistance. Remplacez ?? pour avoir une temporisation de 30s.

```
delay_ms(____); //?? A compléter : Attente de 30s avant la première mesure

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

Lum = read_adc(____); // ? A compléter

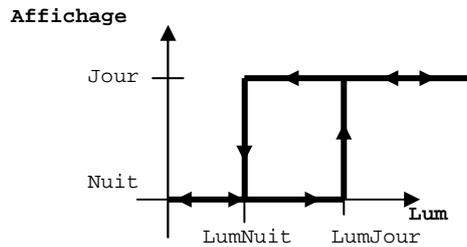
};
```



o **Traitement et affichage de l'éclairement de la photorésistance**

Le programme à réaliser doit afficher « Jour » ou « Nuit » sur le LCD en fonction de l'éclairement reçu par la photorésistance avec une périodicité de 30s.

Le programme à réaliser peut être représenté par le cycle d'hystérésis ci-dessous.



Indications :

➤ Pour l'écriture de cet algorithme (algorigramme), vous disposez des **fonctions suivantes** :

- **Affiche(Texte)** : Affiche le texte contenu dans la variable « Texte »
- **EffaceLCD()**
- **Attente(30s)**

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

Nom	Type	Description
Lum	Octet non signé	Contient une image de la luminosité ambiante.
Texte[]	Tableau de 16 caractères	Contient le texte à envoyer à l'écran (« Jour » ou « Nuit »)

➤ Vous pouvez utiliser une structure alternative (si alors ... sinon, selon ...).

Q8) Etablissez l'algorithme ou l'algorigramme correspondant au cycle d'hystérésis ci-dessus.

Algorithme (ou algorigramme)

Traduction de l'algorithme (algorithme) en langage C (CVAVR)

Algorithmique	Langage C (CVAVR)
Affiche(Texte)	lcd_puts(display_buffer);
EffaceLCD()	lcd_clear();
Attente()	delay_ms() ;
Texte <- "Jour"	sprintf(display_buffer,"Jour");

Q9) Traduisez votre algorithme en langage C et **complétez** le fichier source, partie ci-dessous :

```
lcd_clear();
sprintf(display_buffer,"PPE SERRE (Lum) ");
lcd_puts(display_buffer);
delay_ms(____);///??Attente de 30s avant la première mesure

while (1)
{
// ----- Lecture de l'éclairement reçu par la photorésistance -----
Lum = read_adc(____);          // ? A compléter

// ----- Traitement -----
// A compléter

// ----- Affichage -----
delay_ms(____);///??Attente de 30s avant la mesure suivante
};
```

(3) Partie déclarative du programme à réaliser

Le programme ci-dessus utilise deux types de variable :

- Lum : entier non signé de type octet,
- display_buffer : tableaux de caractères

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

Le programme ci-dessus utilise deux constantes :

- LumJour et LumNuit

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

Q10) Complétez le fichier source C, partie 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 Build"

-> Cochez "Program the Chip"

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



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



D) Test du programme

Appel prof

Pour faire vérifier le fonctionnement du programme

E) Compte rendu

Etablissez un compte rendu destiné à votre dossier final. Précisez les parties de ce document que vous pouvez exploiter pour expliquer la détection et l'affichage du « Jour » et de la « Nuit ».

Annexe 1 : Documentation de la LDR NSL-19M51

Symbol	Parameter	Min	Typ	Max.	Units	Test Conditions
R_L	Light Resistance	20		100	$K\Omega$	10 lux
			5		$K\Omega$	100 lux
R_D	Dark Resistance	20			$M\Omega$	10 sec after removal of light

Annexe 2 : Câblage de la carte « Capteurs » à la carte SSI V1

Les cavaliers correspondant aux entrées des capteurs doivent être positionnés en 1-2.

