




| | | | | |
|--|--|---|---|---|
| Fiche élève : conception du Soft | TS SI |  | P.P.E. Eclairage de scène et commande DMX512 |  |
|  | Logiciel de l'analyseur de trames DMX512 | | | |

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

Objectif

Réaliser les fonctions d'affichage du logiciel en mode « Récepteur » et les implanter dans un microcontrôleur ATMEL 8 bits.

Matériels

PC + Carte ATMEL SSI_V2 + Carte Extension DMX + Alimentation 10V + Contrôleur USB/DMX

Logiciels

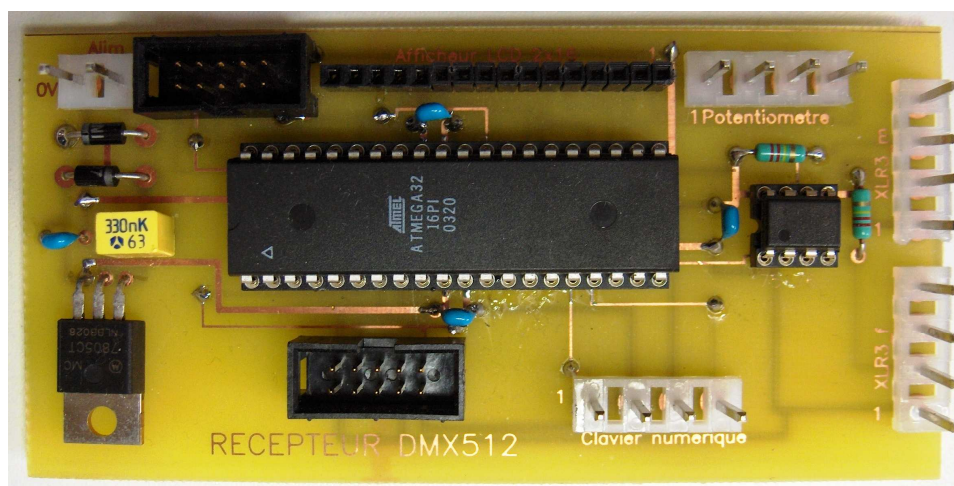
CodeVision AVR. DMX Démo.

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

Fichiers CVAVR élève + Exemple de sniffer DMX512 « MicroScope3a »

A) Présentation

Le logiciel à développer sera implanté dans le microcontrôleur de la carte « Analyseur de trames DMX512 » représenté ci-dessous.



L'utilisateur communique avec l'appareil par l'intermédiaire d'un clavier et d'un afficheur LCD. On accède aux différentes fonctions (émission, réception, ...) par l'intermédiaire de menus.

Actuellement, le logiciel existant propose uniquement l'affichage de 24 canaux sous la forme d'un « Bargraphe ».

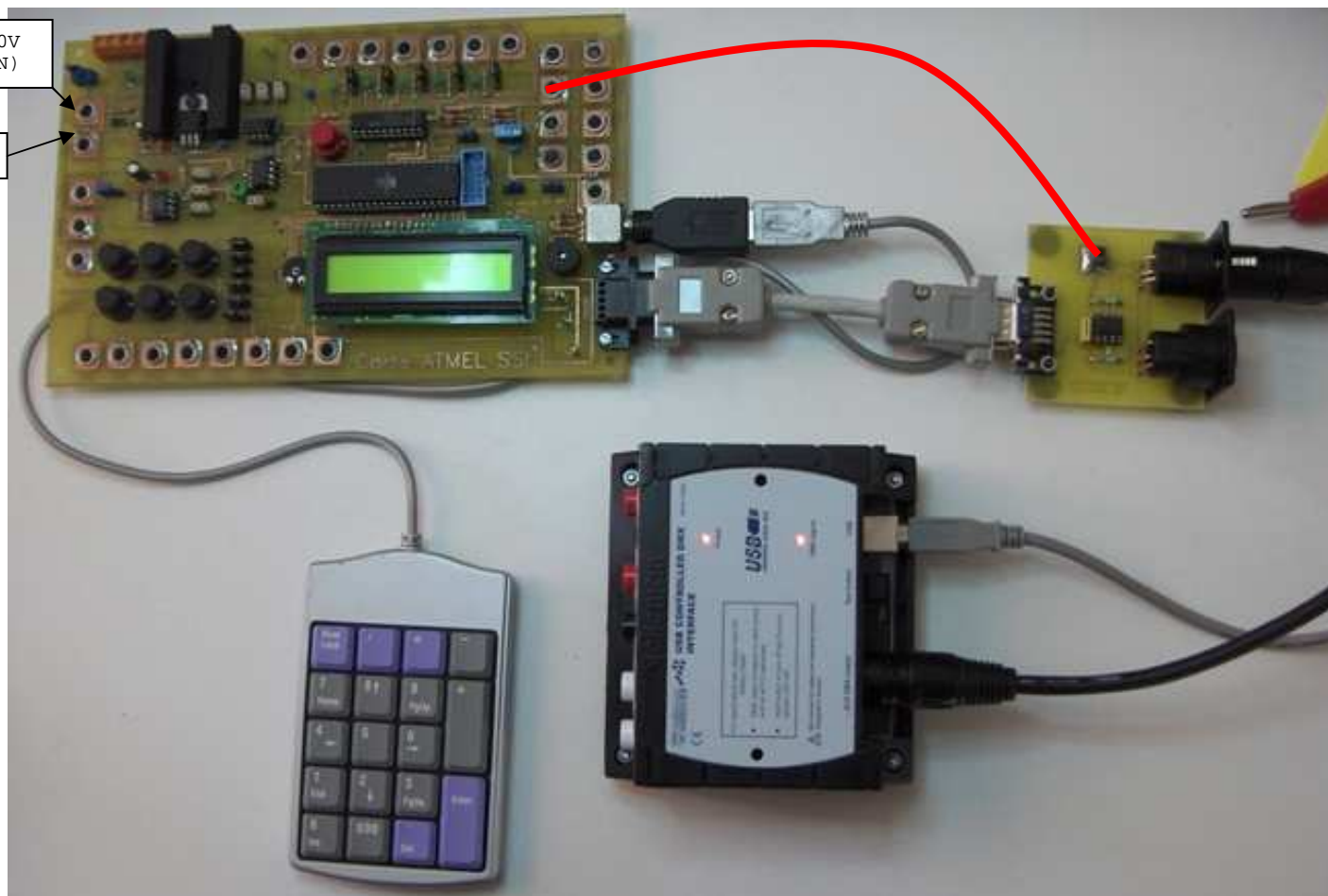
Après avoir analysé ce mode d'affichage, vous réaliserez la présentation des données en hexadécimal, en binaire et en décimal.

| | | |
|----------------------------------|--|---|
| Fiche conception software DMX512 | Logiciel de l'analyseur de trames DMX512 | 1 |
|----------------------------------|--|---|

B) Travail demandé

La mise au point du logiciel à développer sera faite avec la carte SSI_V2 et le contrôleur USB/DMX. Le montage à réaliser est représenté ci-dessous.

Câblage à réaliser



La génération de trames DMX512 sera faite avec le logiciel « **Dmx_demo** ».

B1) Essai du logiciel existant

(1) Transférez le fichier « Recept2.rom », situé dans le répertoire <voir prof>, dans l'ATMEGA32 de la carte SSI_V2 avec le logiciel CodeVisionAvr.

On donne en annexe 2 le graphe des transitions du logiciel existant.

(2) Sélectionnez le mode « Récepteur » puis « Regl. Canal ». Réglez le canal de base avec une valeur n comprise entre 1 et 488. Sélectionnez le mode « Bargraphe »

(3) Lancez le logiciel DMX_Demo
Modifiez les valeurs des canaux DMX n à n+24 pour visualiser le fonctionnement du logiciel.

B2) Analyse de l'affichage en mode « bargraphe »

L'objectif de cette étude est de réaliser une partie de la fonction affichage en mode « bargraphe ». A la fin de cette étude, vous serez capable de réaliser et d'afficher des caractères personnalisés sur un LCD type RTC.

B21) Programmation des caractères d'un bargraphe dans la CGRAM de l'afficheur

On souhaite afficher vingt quatre des cinq cent douze canaux DMX sous la forme d'un bargraphe tel que celui développé dans le « Micro scope 3a» (voir page 16 et 17 du document exemple de sniffer DMX placé sur le site).

Un exemple d'affichage est donné ci-dessous :



Des informations sur l'afficheur LCD sont données dans le document « Datasheets » mis en ligne. La description détaillée de la génération des caractères personnalisés pourra être consultée annexe 1 ou sur le site ci-dessous :

<http://www.chez.com/gdaquino/MC/Afficheur/affich.html>

• Détermination des codes nécessaires au dessin d'un caractère

Q1) Etablissez la liste des valeurs hexadécimales nécessaires au dessin des différents symboles qui apparaissent sur l'écran ci-dessus.
Remarque : Certains de ces symboles existent déjà dans la mémoire de l'afficheur.

• Implantation de la liste des codes dans le texte du programme

La zone des codes à implanter dans la CGRAM est repérée par les commentaires ci-dessous dans le programme Recepte2.c

```
// Constantes en flash  
//-----  
// Caractères personnalisés placés dans la mémoire CGRAM du LCD
```

Q2) Ouvrez le projet Receptel.prj situé dans le répertoire <voir prof> et complétez la partie identifiée « à compléter » avec vos codes.

- **Programmation des codes dans la mémoire CGRAM de l'afficheur**

Les codes des caractères personnalisés doivent être chargés dans la CGRAM de l'afficheur lors du lancement du programme. La partie du programme assurant le chargement est identifiée par les commentaires ci-dessous :

```
// Chargement initial de la mémoire CGRAM de l'afficheur avec les
// caractères personnalisés
```

Q3) En suivant l'exemple donné pour le premier caractère complétez la partie identifiée « à compléter ».

B22 Affichage du bargraphe

La partie du programme assurant l'affichage du bargraphe est donnée ci-dessous.

```
for (i=0;i<24;i++)
{
    Nb_barre = (float)(Table_Donnee[i])/28.33;
    if (i<12) lcd_gotoxy(i+4,0);
        else lcd_gotoxy(i-8,1);
    if (Nb_barre == 0) lcd_putchar(Caractere_Vide);
        else if (Nb_barre == 8) lcd_putchar(Caractere_Plein);
            else lcd_putchar(Nb_barre);
}
```

}

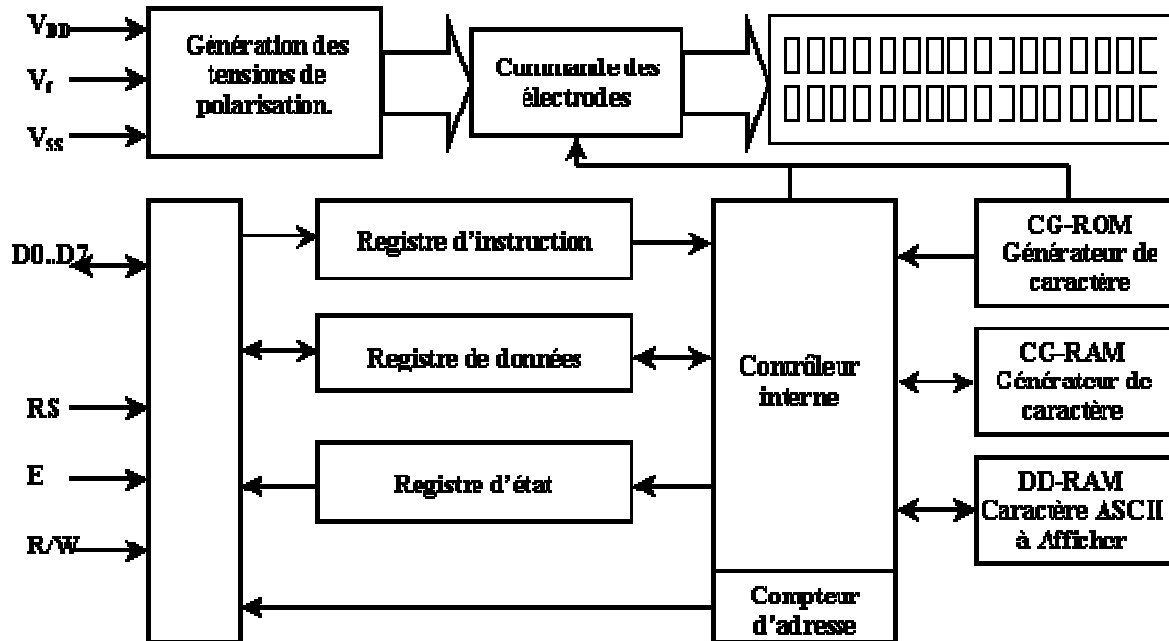
Q4) Dessinez l'algorithme correspondant à cette partie de programme.
Expliquez son fonctionnement.

Algorithme

B3 Présentation des données au formats hexadécimal, binaire et décimal

A terminer

SCHEMA FONCTIONNEL INTERNE



L'utilisateur a accès au :

Registre de données : bidirectionnel, il reçoit les codes ASCII des caractères à afficher. Les codes sont ensuite stockés dans la DD-RAM, d'une capacité de 80 octets. Le registre de données permet aussi de lire le contenu de la DD-RAM.

Registre de contrôle : destiné à recevoir les consignes de contrôle, comme effacer l'afficheur, déplacer le curseur, etc... (accessible uniquement en écriture).

Registre d'état : destiné à indiquer à l'utilisateur si le processeur interne est prêt à recevoir une instruction (accès en lecture seule).

On trouve **deux mémoires CG** (Caractere Generate) :

La CG-ROM : c'est une ROM génératrice de caractères qui fournit 192 motifs différents en matrices de 5 x 7 points. La relation entre le code à transmettre et le motif du caractère est donné dans la documentation de l'afficheur.

La CG-RAM : c'est une RAM génératrice de caractères programmables qui fournit 8 motifs. On peut ainsi obtenir un jeu de 8 caractères personnalisés.

La DD-RAM reçoit les codes des caractères à afficher. L'adresse à laquelle est placé un code dans la DD RAM définit la position du caractère sur le panneau d'affichage. Le pointage d'un élément dans la DD-RAM ou la CG-RAM est déterminé par un compteur d'adresses interne, accessible en lecture.

Cas d'un afficheur a deux lignes de caractères :

Les adresses de la seconde ligne ne sont pas consécutives à celles de la première. Quand l'affichage est décalé, les adresses se décalent uniquement vers la droite ou vers la gauche d'une même ligne. Les adresses assignées à chacune des lignes sont les suivantes :

Ligne n°1 : \$00

Ligne n°2 : \$40

GENERATION D'UN CARACTERE PERSONALISE

La CG-RAM est une mémoire interne du module d'affichage. Sa capacité de 64 octets offre 8 matrices de 5x8 points pouvant être programmées par l'utilisateur. L'espace adressable donnant accès à ces caractères (DD-RAM) va de l'adresse \$00 à \$07.

Programmation d'un caractère

Chaque caractère occupe 8 octets. Un pixel devant apparaître en noir sera positionné à 1.

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Code Hex |
|----------------|---|---|---|---|---|---|---|----------|
| Non utilisé | | | | | | | | \$1F |
| | | | | | | | | \$03 |
| | | | | | | | | \$03 |
| | | | | | | | | \$07 |
| | | | | | | | | \$0D |
| | | | | | | | | \$19 |
| | | | | | | | | \$10 |

Ainsi pour programmer le caractère ci-dessus il faudra écrire le code, \$1F à l'adresse \$40, le code \$03 à l'adresse\$41,etc ..., et le code \$10 à l'adresse \$47. Le caractère suivant occupera les adresses CG-RAM \$48 a l'adresse \$4F et ainsi de suite.

| Code caractère DD-RAM | Adresse CG-RAM à programmer |
|--------------------------|--------------------------------|
| \$00 | \$40 à \$ 47 |
| \$01 | \$48 à \$ 4F |
| \$02 | \$50 à \$ 57 |
| \$03 | \$58 à \$ 5F |
| \$04 | \$60 à \$ 67 |
| \$05 | \$68 à \$ 6F |
| \$06 | \$70 à \$ 77 |
| \$07 | \$78 à \$ 7F |

