

1 Introduction

Ce document sert d'explication au fonctionnement de la rampes à LED.

2 Matériel

2.1 Control

2.2 Drivers

3 Soft

Cette partie expolique le fonctionnement du soft de la rampes à LED.

3.1 Les états

Chaqu'une des 10 sorties de la carte de controle et branché avec un ou plusieurs drivers pour controlé grace au MOSFET présent sur le drivers. On distingue 4 états pour les sorties.

- 0 : Eteinte
- 1 : En Cours d'allumage
- 2 : Allumé
- 3 : En cours d'extinxion

Chaqu'un des états si desus dure 1h. Les états 0 et 2 peuvents être renouvelé.

La valeurs de chaque sortie correspond à l'état mais aussi a la duré en minute dans cette états.

L'état eteint la sortie est toujours égale a 0 donc le ou les MOSFET branché sur cette sortie sont non passant. L'état allumé la sortie est toujours égale a 1 donc le ou les MOSFET branché sur cette sortie sont passant. L'etat en cours d'allumage la sortie est est variable avec un PWM avec un duty cycle variable en fonction du temps. On incremente toute les 3 min de 5% le duty cycle. L'état en cours d'extixion est le contraire de l'état en cours d'allumage. On démarre avec un duty cycle a 100% pour progressivement arrivé a 0%.

Les états de chaque sortie pour chaque heure de la journée sont stocké dans un tableau dans l'EEPROM. Le tableau est de la forme suivante:

Chaque case du tableau fait 1 octet est contient l'états de la sortie à une certaine heure de la journée.

3.2 L'algorithme général du programme

Le programme fonctionnera avec 3 timers:

- timer 1 : Se timer est réglé sur 1 second il sera toujours actif. Toute les seconds le programme ira lire l'heure sur la RTC, l'ecrira dans l'eprom, testera si on a changé d'heure (ex:10H -> 11H). Si une des sortie est dans un des 2 états EnCours le timers 1 testera si 3 min ce sont écoulé pour l'augmentation du timer 3.
- timer 2 : Ce timer est réglé toute les 20ms il est fonctionel que si une des sortie est dans un des 2 états EnCours. Il sert a allumé les sortie qui sont dans l'état en cours d'allumage, et eteindre celle qui sont dans l'état en cours d'extinction. Il demare aussi le timer 3 avec le bon nombre de milli second.
- timer 3 : Ce timer est réglé en fonction des minutes $\lfloor (minute/3) \rfloor / 1000$ ce qui donne un nombre de miliseconde entre 1 et 20 ms. Sont action sera en opposition avec le timer 2 etteindre les sortie qui sont dans l'état en cours d'allumage et allumé celle qui sont dans l'état en cours d'extinction.

Algorithme principal: **Chaque fonction sera détaillé apres**

```

global ??? heure = lireHeure(); //on initialise l'heure grace a la
    fonction de lecture de l'heure sur la RTC.
global char nouvelleHeure = 1; // Variable qui sert a savoir si on est
    ..... passe a une nouvelle heure. Initialise a 1 car au demarage du
    ..... programme rien est initialise.
global char change = 0; ..... // Variable qui sert a savoir si une des
    ..... sortie est dans un tat non stable.
global char tableSortie[10]; ..... // Variable qui sert a savoir quelle sont
    ..... les tats des sortie.
global char 3min = 0; ..... // Variable qui sert a savoir si on a pa
    ..... 3min depuis la derniere fois que l'on a augm nt le timer 3
global char Xms = 0; ..... // Variable qui sert au temps du timer 3

demarreTimer1();
while(1){
    if (nouvelleHeure) {
        nouvelleHeure = 0;
        Xms = 0;
        lireSortie(heure, tableSortie);
        change = change(tableSortie);
        if (change)
            demarreTimer2();
        else
            eteindreTimer2();
    }
    if (change){
        if (3min)
            Xms = Xms + 1 ;
    }
}

```

```
}  
sleep(65); // On dort 65 ms pour pas toujours etre actif
```