

Mini Projet : Réalisation d'un système temps réel.

CASIER DE RANGEMENT DES RESISTANCES INTELLIGENT

Le but de ce TP est de réaliser un système d'assistance pour gérer les résistances dans les casiers qu'on trouve classiquement en salle de TP (meuble casier de 5 colonnes de 12 cases pour les 60 résistances de la série E12 de 10 Ohm jusqu'à 820 kOhm). Grâce à ce système, on pourra tester des résistances et allumer le casier où les ranger, et commander les résistances dont les casiers sont vides en envoyant un mail.

Sur le mur, intégré à une planche support derrière chaque casier, il y aura une led neopixel, ainsi qu'un capteur ils (interrupteur sensible à l'aimant fixé au casier translucide devant les leds et le capteur). On pourra ainsi éclairer un casier ou plusieurs casiers parmi les 60 en RVB (à priori seuls Rouge et Vert nous intéresse et, et savoir quels casiers sont ouverts ou fermés grâce aux capteurs.

L'interface utilisateur au niveau microcontrôleur comportera :

- Un bouton BP_DEMANDE_TEST et un connecteur deux points pour enficher la résistance à tester.
- Un bouton BP_DEMANDE_APPRO et un lecteur Ibutton one_wire (patte E/S ONE_WIRE)
- Une sortie Haut parleur pour diffuser des messages à l'utilisateur

L'utilisateur enclipsa la résistance à tester/vérifier sur un connecteur 2 points puis appuiera sur le bouton pour lancer un test de valeur : BP_DEMANDE_TEST. Le casier où ranger la résistance s'illuminera tant que la résistance sera présente pour un maximum de 5 secondes.

L'utilisateur pourra demander le réapprovisionnement des casiers en activant le mode réapprovisionnement pour visualiser en rouge les casiers ouverts (casiers déclarés à réapprovisionner) et en vert les casiers fermés.

Passer son badge Ibutton déclenchera alors l'envoi d'un mail de demande de réapprovisionnement

1 Gestion de l'éclairage des casiers :

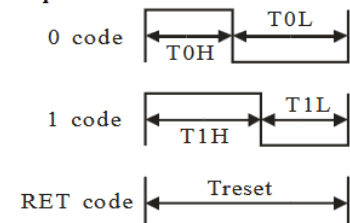
L'état des couleurs à afficher sur les 60 leds derrière les casiers sera stocké dans un tableau d'entiers 32 bits

L'allumage des leds Néopixels nécessite des chronogrammes très précis, on les générera à l'aide d'un PWM dont on fera varier le rapport cyclique après avoir ajusté la période à 1,4 us (on rallonge le timing au max) :

Data transfer time ($T_H + T_L = 1.25\mu s \pm 150ns$)

T0H	0 code ,high voltage time	0.35us	$\pm 150ns$
T1H	1 code ,high voltage time	0.9us	$\pm 150ns$
T0L	0 code , low voltage time	0.9us	$\pm 150ns$
T1L	1 code ,low voltage time	0.35us	$\pm 150ns$
RES	low voltage time	Above 50μs	

Sequence chart:



Un code 1 sera envoyé en réglant le rapport cyclique à 70 % pendant une période de PWM

Un code 0 sera envoyé en réglant le rapport cyclique à 30 % pendant une période de PWM

Un reset sera envoyé en réglant le rapport cyclique à 0 % pendant 40 périodes de PWM ($40 * 1.4\mu s > 50\mu s$)

La fonction gérant la séquence postule que quelqu'un d'autre gère la mise à jour du tableau.

On provoquera soit un transfert DMA pour mettre à jour la valeur du PWM, soit une IT à chaque cycle PWM

2 Gestion des fonctions mesure et réapprovisionnement

On écrira une tâche qui scruptera 10 fois par seconde l'appui sur l'un des boutons « BP_DEMANDE_TEST » ou « BP_DEMANDE_APPRO ». Comme l'un et l'autre des boutons est amené à gérer l'affichage sur les casiers, il ne faudra pas que les deux tâches gérant la mesure et la scrutation des ouvertures fermetures des casiers soient éveillées en même temps. La scrutation des boutons s'arrête quand on quitte le MODE_EFFACE

Pour la mise à jour du tableau de consigne des leds, on a besoin d'implémenter trois types de mise à jour :

- effacement du tableau pour éteindre toutes les leds : **MODE_EFFACE**
dans ce mode on scrute les boutons, et si on voit un bouton enfoncé ,
on bascule en mode **MODE_TEST** si **BP_DEMANDE_TEST** est enfoncé (forçage !),
ou parce que quelqu'un a inséré une résistance de valeur différente de la valeur précédente mesurée.
on bascule en mode **MODE_APPRO** si **BP_DEMANDE_APPRO** est enfoncé.

- allumage d'une seule case en bleu et extinction des autres cases : **MODE_TEST**
On va activer 10 fois par seconde la mesure de la resistance
une variable **num_case_resistance** indiquera le casier concerné de 1 à 60
variable disponible quand la mesure est finie (on sera prévenu par une notification)
si la valeur de la variable est négative alors il faut faire clignoter le casier
(on récupère son numéro en prenant la valeur absolu)
si la valeur est nulle alors aucune case ne doit être allumée, cela correspond à resistance absente
ou hors zone (<10 Ohm ou >820KOhm)

- allumage en vert des casiers fermés, allumage en rouge des casiers ouverts : **MODE_APPRO**
un autre tableau donne l'état des reed : **unsigned char etat_reed[60]** ;
ce tableau est mis à jour par une tâche qui se déclenche toutes les 10ms pour scrupter une ligne de contacteurs REED : l'ensemble du tableau est mis à jour toutes les 120ms soit 8 fois par seconde
 - on lit les 5 entrées,
 - on met à jour les 5 cases concernées du tableau, on active une notification s'il y a changement
 - on change de ligne (ligne10, ligne12, ligne15, ligne18, ligne22, ligne27, ligne33 ligne82)
on remet la ligne précédente à 1, on met la suivante à 0
 - on fait un délai de 10ms

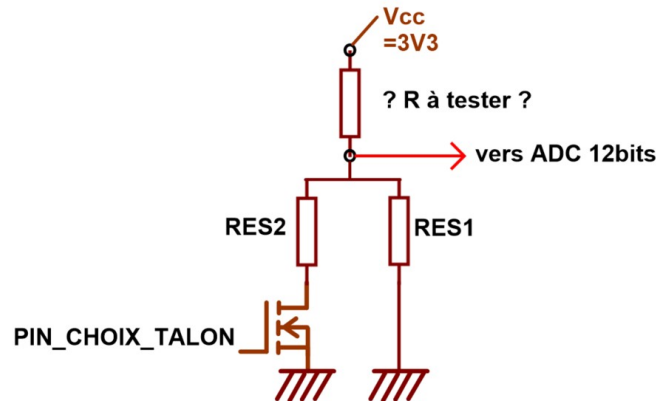
Dans le mode **MODE_TEST**, on y reste 5 secondes au plus, où on le quitte plus tôt si on change l'état du bon casier. Après l'avoir quitté, on y revient pas si quelqu'un a oublié sa résistance sur le testeur sauf si appui BP...

Dans le mode **MODE_APPRO**, on y reste 5 secondes au plus, après le dernier changement de l'état global des tiroirs, ou on le quitte immédiatement si un mail est envoyé suite à la présence d'un badge connu...

3 Gestion de la mesure de la résistance

Cette tâche est déclenchée périodiquement, on devra provoquer une mesure de tension : la résistance inconnue est placée dans un pont diviseur dont la tension de sortie varie en fonction de sa valeur (voir fichier excell joint) qui permettra d'identifier la valeur de la résistance et donc le casier où la ranger. La numérotation des casiers se fera en ordre de résistance croissante.

Pour des raisons de précision de mesure, on ne peut pas travailler avec une seule résistance talon, il en faut deux, on résout cela avec cette électronique :



Une patte en sortie permet de connecter la résistance RES2 à la masse ou pas, changeant la nature du pont diviseur... les calculs théoriques sont donnés dans `calc_tableaux_seuils_comparaisons.xls`

On disposera d'une IT de fin de conversion qu'on devra rendre capable de notifier cette fin de conversion à notre tâche en attente de fin de conversion. Quand on change la commande du transistor, il faut attendre un peu que la nouvelle tension se stabilise.

Par l'analyse de deux tableaux de seuils (un par pont diviseur), on pourra déterminer quel casier allumer.

Parfois une seule mesure ADC suffit, parfois il faut changer de configuration de test

- RES1 seule pour les résistances de forte valeur
- RES1 // RES2 pour les résistances de faible valeur

4 Gestion de la lecture d'un badge lbutton

On tentera de déclencher la lecture du badge 5 fois par seconde, on transformera le code bloquant mis à disposition en un code non bloquant en utilisant une notification par l'IT Timer qui détecte la fin d'un motif.

Si un badge vient d'apparaître, on interrogera par liaison série une base de donnée qui dira si le badge est connu ou pas... Si on est en mode **MODE_APPRO**, il faut alors envoyer un message série pour donner la matrice de réapprovisionnement à faire

5 Gestion du son :

Le son se joue en permanence : soit du silence, soit un vrai son.

Il faudra mettre en place :

une tâche player qui gère la machine à état du son

une tâche de lecture SPI pour charger des buffers audio (rempli avec du silence ou des data de vrai son),

une tâche de mise à jour du DAC qui vide les buffer en permanence.

