Centre professionnel du Nord vaudois
Filière d'automatique et d'électronique
Modules complémentaires techniques

CPNV

Chronomètre

P1704_Manettes

SANDOZ Pierre-Olivier
PICOTTE ALEXANDRE
25/05/2018

Introduction :

Dans le cadre de notre projet, nous avons reçu une commande du groupe chevaliers et dragon, ces derniers avaient besoin de deux manettes, une pour chaque chevalier. Une des exigences de cette commande était l'implémentation et l'affichage d'un chronomètre.

Deux problèmes sont survenus lors de la programmation de la manette, un problème pour afficher le temps restant et un problème pour le calcul du temps restant.

Affichage :

Le premier souci était l'affichage du chronomètre, la bibliothèque nécessaire au fonctionnement des écrans TFT lcd ne propose que 3 manières pour afficher des éléments sur l'écran : le chargement d'image, la création de lignes ainsi que de rectangles et pour finir l'écriture de texte. Nous avons choisi l'écriture de texte car charger des images ralentit le programme et l'utilisation de lignes pour « dessiner » des chiffres auraient été longs et fastidieux. Par contre avec le choix du texte apparaissait un nouveau problème, la variable du texte à afficher est une chaine de caractère et la fonction millis() retourne un *unsigned long*, il faut donc trouver un moyen de convertir la valeur du chronomètre en chaine de caractère. Nous première tentative, que nous avons rapidement avorté, était la mise en place d'un switch case à 10 éléments pour trois variables : les secondes, les dizaines de secondes et les minutes. Chaque variable suivant sa valeur se voit attribuer un caractère qui est ensuite affiché sur l'écran. Or cela aurait été long à implémenté, donc nous avons opté pour convertir les secondes et les minutes en chaine de caractère.

```
/*************************************************************************
   Function Name: conversion
   Description:  convert an integer into a character array,
                 then display it on the screen
   Parameters: the integer to convert, the position X on the screen and the color
   Return: nothing
*************************************************************************/

void conversion(int toConvert, int place, int color) {
  char b[6];
  String str;
  str = String(toConvert);
  str.toCharArray(b, 6);
  drawText(1, color, color, color, place, 8, b);
}
```

Cette fonction est prise d'un site (http://www.instructables.com/id/Converting-integer-to-character/) Cette fonction permet de convertir les int en chaine de caractère puis de les afficher là où l'on souhaite sur l'écran. La fonction « drawText() » est reprise du code de l'ancienne manette faite par l'ancien groupe « chevaliers et dragon », cette fonction permet de choisir la taille, la couleur, l'emplacement et le texte à écrire. Il est important que le tableau de caractère ait un emplacement vide, il permet de signaler à l'arduino quand se termine la string.

Compteur :

La fonction chronomètre est passé par trois versions, qui approchait le problème selon différents angles, la version finale possède des éléments de la première version et de la deuxième version.

Première version :

La première version était sous forme de décompteur. Lors de l'initialisation du mode de jeu 3, appelé HUD, l'arduino mettait la variable minute à 3 et la variable seconde à 0, puis lorsque l'arduino exécutait sa fonction chronomètre, il vérifiait si la différence entre millis et la dernière mesure du chronomètre était supérieur ou égal à 1000 ; si oui alors il diminuait la variable seconde de 1 et si seconde devenait négatif alors minute était réduite de 1, lorsque le décompteur atteignait 0 minute et -1 seconde alors l'arduino lancait l'écran de game over. Le problème majeur de cette première version était son décompte faussé, après test il mettait environ 1m30 pour décompter 1min. La première version affichait aussi faux le temps restant, cela était dû à un oubli de laisser un espace vide pour le tableau de caractère.

```
timer = millis()
if (timer - last_timer >= 1000) {
  last_timer = timer;
  seconde--;
  if(seconde <0){
    minute--;
    if(minute<0){
      mode=5; // mode pour le game over
    }
    seconde=59;
  }
}
```

Pour essayer de rectifié les erreurs du chronomètre nous avons inclut dans la fonction la ligne « timer=millis() », car avant il mettait à jour la variable timer qu'une fois dans la loop, mais cela n'a rien changé.

Deuxième version :

Pour cette version nous avons décidé de reprendre depuis la base, nous avons changé le décompteur par un simple chronomètre qui lorsqu'il atteint 3 min arrête la partie.

```
timer = millis()
if (timer - timer_debut_jeu >= temps_de_jeu/*180000*/) {
  mode=5; // mode pour le game over
```

Nous avons mis en place ce simple chronomètre pour avoir une sécurité supplémentaire et vérifier si le code n'était pas trop lourd et ralentissait trop le programme. Après une petite correction, car nous avions initialisé « temps_de_jeu » par un int, ce qui le faisait overflow lorsque nous essayions de lui assigner 180'000. Les tests ont été concluants, le game over arrive lorsque la partie atteint les trois minutes. Nous sommes donc partis de ce chronomètre qui compte correctement pour ensuite en extraire les secondes et minutes écoulés.

```
double minute, seconde;
int minute_int, seconde_int;
unsigned long ecouler
minute = int((timer - timer_debut_jeu) / 60000);
ecouler = (timer - timer_debut_jeu) % 60000;
seconde = int(ecouler / 1000);
seconde_int = int(seconde);
minute_int = int(minute);
```

Ce code permet de lire les minutes et les secondes écoulés, la transition de double à int est nécessaire pour éviter d'afficher les zéros après la virgule. Ensuite pur des soucis de rapidité et de clarté, nous avons remis un compteur similaire à la première version qui chaque seconde affichera les minutes et secondes écoulés. Malheureusement cette version comptait de 1,1s en 1,1s.

Troisième version :

La troisième version résulte du changement dont l'arduino compte les secondes. Avant nous faisions une comparaison entre deux timer, lorsque leur différence était égale ou supérieur à 1000 pour 1s, le programme faisait les actions nécessaires pour afficher les secondes et minutes et on mettait à jour le deuxième timer (comme sur la première version). Mais avec cette méthode le compteur est toujours décalé, il n'enregistre pas le décalage qu'il subit. Alors nous avons changé par une comparaison entre le timer et le timer du début de partie dont on additionne un écart qui est égale à 1000, chaque fois que le compteur enregistre une seconde il fait les actions nécessaires et incrémente l'écart de 1000, ainsi on garde l'erreur en mémoire et le chronomètre ne se décale plus. En combinant cette version avec le décompteur de la première version et le chrono supplémentaire de la deuxième version on obtient la version finale.

```
void chronometre() {
  chrono = millis();
  if (chrono /*- lastChrono*/ >= ecart + timer_game) {
    ecart = ecart + 1000;
    //lastChrono = chrono;
    Serial.println(ecart);
    conversion(seconde, 139, 0);
    seconde--;
    if (seconde < 0) {
      seconde = 59;
      conversion(minute, 121, 0);
      minute--;
      conversion(minute, 121, 255);
    }
    conversion(seconde, 139, 255);
  }
  if (chrono - timer_game >= temps) {
    fin = true;
  }
```