*Murillo Arthur | TD2*

DUT Informatique S4

30/03/2020

**M41 01C |PEL - Systèmes temps-réels : Examen**

***Question 1***

On utilisera le port B pour contrôler tous les périphériques. En effet, il a été décidé que tous les périphériques seraient branchés sur ce port. De plus, on utilisera le timer 0 pour mesurer un délai de 10 secondes pour savoir quand mesurer. Enfin on utilisera le convertisseur Analogique Numérique pour pouvoir connaître la tension mesurée par le récepteur infra-rouge.

***Question 2***

void init() {

allumerPeriph(PortB) ;

allumerPeriph(ConvAN) ;

allumerPeriph(Timer0) ;

timerModeDelai(Timer0, HDIV2, 420000000UL, REPETITIF, INC) ;

autoriserITTimer(Timer0, LIMITE, 1, interruption\_detecFumee) ;

programmerLigne(PortB, 12, ENTREE) ;

initConvAN(10, RES10BITS) ;

choisirEntreeConvAN(8) ;

declenchementConvAN(PROG) ;

programmerLigne(PortB, 13, SORTIE0) ;

programmerLigne(PortB, 14, SORTIE0) ;

programmerLigne(PortB, 15, ENTREE) ;

activerFiltrageLigne(PortB, 15, 20) ;

}

***Question 3***

void interruption\_detecFumee() {

ecrireLigne(PortB, 13, 1) ; // On allume l’émetteur

lancerConversionAN() ;

while ( !testerEtatConvAN(3)) {} // On attend la fin de la conversion

float tensionMesuree = lireTensionConvAN(8) ;

if (tensionMesuree > 0,2) {

ecrireLigne(PortB, 14, 1) ; // On allume l’alarme si la fumée est détectée

}

ecrireLigne(PortB, 13, 0) ; // On éteint l’émetteur

}

***Question 4***

int main () {

initCarte() ;

init() ;

lancerTimer(Timer0) ;

while (true) {

while (lireLigne(PortB, 15) == 0) {} // On attend un appui sur le bouton

if(lireLigne(PortB, 14) == 1 ) { // Si l’alarme est allumée

ecrireLigne(PortB, 14, 0) ; //On l’éteint

while (lireLigne(PortB, 15) == 1) {} /\* On attend que le bouton soit totalement relaché pour ne pas refaire un tour de boucle trop vite \*/

}

else { //Si l’alarme était éteinte, on veut la tester

ecrireLigne(PortB, 14, 1) ; //On l’allume

while (lireLigne(PortB, 15) == 1) //On attend que le bouton soit relâché

ecrireLigne(PortB, 14, 0) ; //On l’éteint

}

}

return 0 ;

}