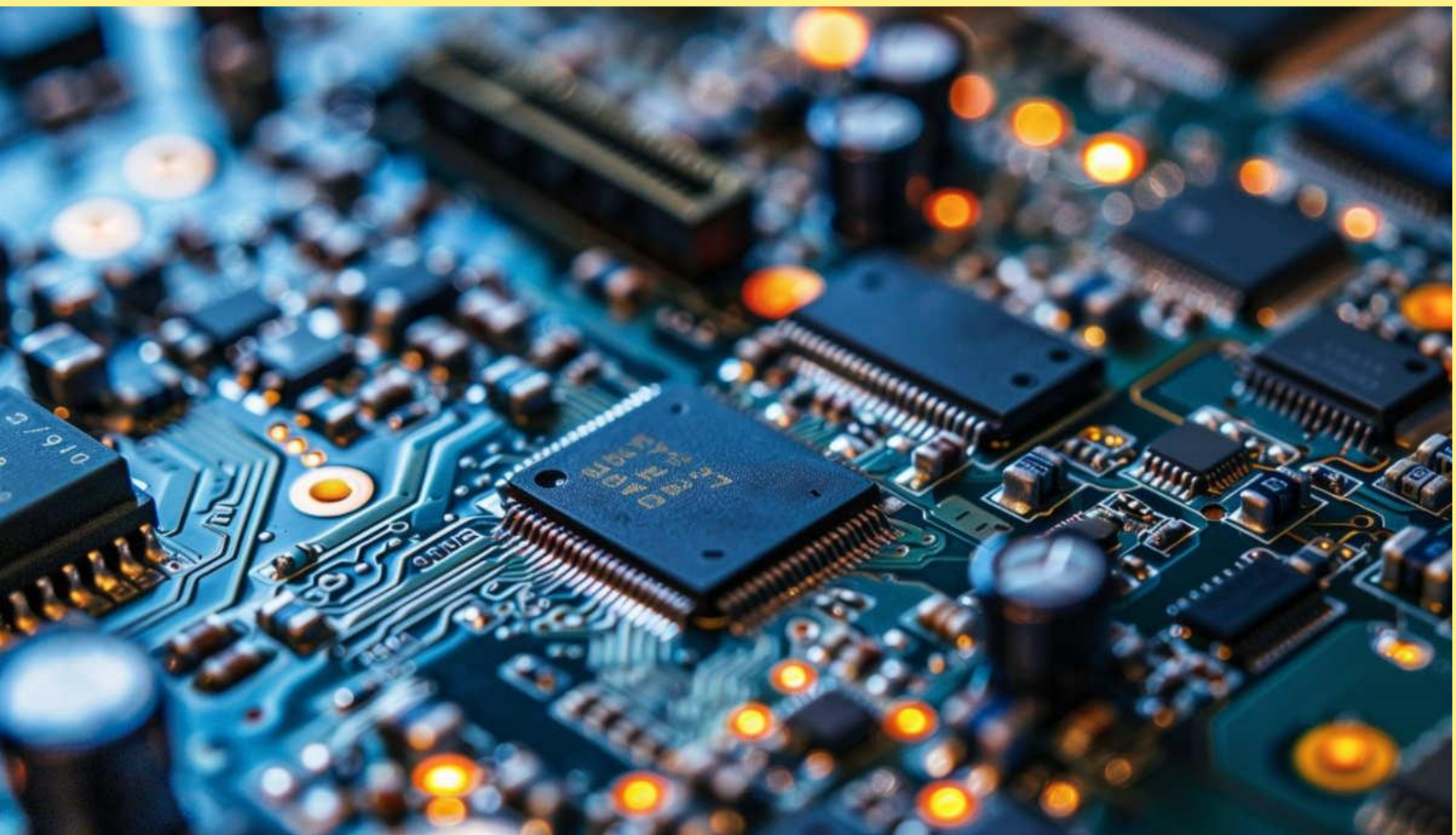


CPI A2 – PROJET SYSTEMES EMBARQUÉS

PROSIT 6 -INTERROMPEZ MOI



Contexte :

Dans le cadre du projet THS, l'équipe utilise un Arduino pour collecter des données de capteurs (RTC, GPS, luminosité) et les enregistrer sur une carte SD. Cependant, il est nécessaire de pouvoir basculer entre différents modes de fonctionnement en fonction de la durée de pression d'un bouton poussoir.

Les devs ont relevé le problème que lorsque qu'on appuie sur un bouton

Mots inconnus :

L'horloge RTC : Une horloge temps réel ou HTR (en anglais, real-time clock ou RTC) est une horloge permettant un décompte très précis du temps (par exemple en nanosecondes)

Problématique :

Comment, à l'aide des interruptions pouvons-nous traiter le fait de changer de mode et avoir une acquisition de donnée ?

Plan D'action :

Comment fonctionne les interruptions [30mins]

Comprendre le code qui nous est donné et l'améliorer[60mins]

Implémenter le code en Arduino [90mins]

Réalisation :

```

1  #define LEDPIN 7
2  #define PUSHPIN 8
3  int randomNumber = 0;
4  int data;
5  const int led = 9;
6  const int inter = 2;
7  volatile bool bascule = false;
8  void acquisition(){
9      randomNumber = random(3, 7); // Je tire un nombre au hasard entre 3 et 6
10     Serial.println(randomNumber); // delai du nombre de secondes simulant l'acquisition de données
11     //digitalWrite(led, LOW); // Eteindre la LED après acquisition
12 }
13 void basculement(){
14     bascule = !bascule;
15     Serial.println("interruption");
16     if (bascule)
17     {
18         digitalWrite(led, HIGH);
19     }
20     else
21     {
22         digitalWrite(led, LOW);
23     }
24 }
25 void setup(){
26     randomSeed(analogRead(0)); // Initialisation générateur de nombres aléatoires
27     pinMode(LEDPIN, OUTPUT); // Initialisation LED
28     pinMode(PUSHPIN, INPUT); // Initialisation bouton
29     Serial.begin(9600);
30     Serial.println(" ");
31     Serial.println("Demarrage");
32     pinMode(led, OUTPUT);
33     pinMode(inter, INPUT_PULLUP);
34     attachInterrupt(digitalPinToInterrupt(inter), basculement, FALLING); // Activer l'interruption sur la broche inter
35 }
36 void loop(){
37     // Le code principal est géré dans l'interruption
38     delay(randomNumber * 1000);
39     acquisition();
40 }

```