

Projet 2

SNIOT

Téo Champion

SN 2A – SeaTech

Introduction

L'objectif est de faire fonctionner un capteur PPG sur un embarquée (ESP32) sous Zephyr OS.

Ce capteur à pour but de mesurer la fréquence cardiaque avec un signal infrarouge, les traiter en temps réel (threads).

Le projet permet aussi d'apprendre la programmation bas niveau de périphériques via I²C, la gestion des capteurs sous Zephyr, et la planification de tâches temps réel.

1.1 Analyse et architecture du système

1.

Le mot-clé volatile indique au compilateur qu'une variable peut être modifiée de manière externe (thread par exemple).

2.

- k_mutex : empêche deux threads d'accéder à une ressource en même temps.
- k_sem : utile pour synchroniser une tâche avec un événement (ex. nouvelle donnée capteur).

3.

Tâche

- T1 : Initialisation du capteur MAX30102
- T2 : Acquisition du signal PPG via interruption
- T3 : Traitement du signal (FFT ou détection de pics)
- T4 : Envoi périodique des données via Wi-Fi
- T5 : Gestion START/STOP acquisition

4.

Capteur (MAX30102) → (I²C) → Thread Acquisition → (données) → Thread Traitement
→ (résultat HR, SpO2) → Thread Wi-Fi

1.2 Capteur MAX30102

5.

PIN	NAME	FUNCTION
1, 5, 6, 7, 8, 14	N.C.	No Connection. Connect to PCB pad for mechanical stability.
2	SCL	I ² C Clock Input
3	SDA	I ² C Data, Bidirectional (Open-Drain)
4	PGND	Power Ground of the LED Driver Blocks
9	V _{LED+}	LED Power Supply (anode connection). Use a bypass capacitor to PGND for best performance.
10	V _{LED+}	
11	V _{DD}	Analog Power Supply Input. Use a bypass capacitor to GND for best performance.
12	GND	Analog Ground
13	INT	Active-Low Interrupt (Open-Drain). Connect to an external voltage with a pullup resistor.

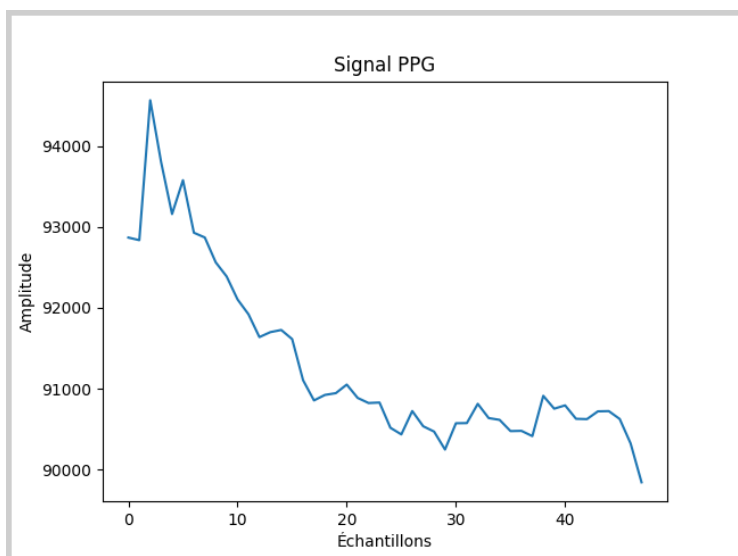
Pour le pin INT il faut rajouter une résistance de pull-up.

6,7,8,9,10.

A rajouter dans le fichier : esp32_devkitc_procpu.overlay :

Voir main.c

prj.conf



```
Code python : import serial

import matplotlib.pyplot as plt

ser = serial.Serial('/dev/ttyUSB0', 115200)
data = []

for i in range(50):
    line = ser.readline().decode(errors="ignore").strip()
    if not line:
        continue
    try:
        r, ir = map(int, line.split())
        data.append(r)
    except ValueError:
        continue

plt.plot(data)
plt.title("Signal PPG")
plt.xlabel("Échantillons")
plt.ylabel("Amplitude")
plt.savefig("ppg_signal.png")
print("Graphe sauvegardé sous ppg_signal.png")
```

1.3 Acquisition et traitement du signal

11.

L'acquisition se fait par : `drdy_trigger_mode(max30102_dev);`

Les données par `sensor_sample_fetch(max30102_dev);`

La fft est donnée par : `static float fft_processing(const struct ppg_sample *segment)`

(Dans `processing.c`)

Voir programme.

12.

La stratégie a été implémentée en intégrant la FFT au flux d'acquisition du capteur MAX30102 : les données PPG sont collectées via interruption, stockées dans un tampon de 1024 échantillons puis traitées par la FFT pour extraire la fréquence dominante et calculer la fréquence cardiaque. Les tests sur carte ESP32 montrent une détection stable autour de 1,2 Hz (~72 bpm) et un temps de calcul d'environ 25 ms, confirmant le bon fonctionnement et la conformité de la solution.