

Documentation Concepteur BertheVarioTac

Projet d'alti vario gps parapente

à base de 2.8inch ESP32-2432S028

Version : 20250124c

<https://github.com/gitberthe/BertheVarioTac>

Compilé avec Visual Studio Code / PlatformIO

Espressif Systems
ESP32 Dev Module
ESP32-2432S028 Arduino
Wroom 32

langage C++.

Table des matières

1 Mise en place de l'environnement logiciel :.....	3
2 Calibration de l'écran :.....	3
3 Connexion Gps GY-NEO6MV2 :.....	4
4 Connexion capteur de pression BMP180 ou MS5611 :.....	4
5 Connexion capteur magnétique QMC5883 :.....	4
6 Connexion de la batterie :.....	4
7 Alimentation :.....	4
8 Réglage du potentiomètre :.....	4
9 haut parleur :.....	5
10 Utilisation du PCB :.....	5
11 Téléchargement du firmware :.....	5
12 Quelques photos :.....	6

1 Mise en place de l'environnement logiciel :

Le logiciel est compilé avec Visual Studio Code / PlatformIO. Après l'ouverture du projet et téléchargement automatique des librairies, il faut décompresser 2.8inch_ESP32-2432S028R.rar.

* Copier alors le fichier

2.8inch_ESP32-2432S028R/1-Demo/Demo_Arduino/7_1_Touch_button_ILI9341_LovyanGFX/TFT_eSPI bottom layer replacement file/User_Setup.h ou ./src/Screen/User_Setup.h dans BertheVarioTacPlatformIO/.pio/libdeps/esp32dev/TFT_eSPI/User_Setup.h.

* De même, il faut copier BertheVarioTacPlatformIO/.pio/libdeps/esp32dev/LovyanGFX/examples/Advanced/LVGL_PlatformIO/src/lv_conf.h dans ./src/Screen/lv_conf.h

* La librairie pour le gestionnaire de fichier Wifi doit être modifiée.

Dans BertheVarioTacPlatformIO/.pio/libdeps/esp32dev/ESP32 File Manager for Generation Klick ESPFMfGK/src/ESPFMfGKGa.cpp" ligne 6 : crc32.h => CRC32.h

La compilation du logiciel peut alors s'effectuer.

2 Calibration de l'écran :

Il faut obtenir 8 paramètres de calibration de l'écran. Pour cela dans src/Screen/CLGFX.cpp décommenter la portion adéquate dans CLGFX::InitScreen() (ligne 81) pour récupérer sur le port série de PlatformIO les 8 paramètres à recopier par la suite dans le tableau de calibration calData[] (ligne 78). On a ainsi les bonnes coordonnées XY d'un point appuyé.

```
70
71
72 //////////////////////////////////////////////////
73 /// \brief Initialise l'ecran et le touch pad
74 void CLGFX::InitScreen()
75 {
76     g_tft.begin();
77     g_tft.setRotation(7);
78     g_tft.setBrightness( g_GlobalVar.m_Config.m_luminosite );
79     uint16_t calData[] = { 317, 3787, 3847, 3813, 331, 232, 3860, 238 };
80
81     /*
82     // procedure de calibration à faire une fois
83     g_tft.calibrateTouch( calData , TFT_WHITE , TFT_BLACK );
84     for ( int ic = 0 ; ic < 8 ; ic++ )
85     {
86         Serial.print( calData[ic] );
87         Serial.print( ", " );
88     }
89     */
90     g_tft.setTouchCalibrate(calData);
91
```

A noter que l'éclairage de l'écran perturbe grandement la mesure de luminosité (à isoler donc).

3 Connexion Gps GY-NEO6MV2 :

Il faut alimenter correctement le module par le connecteur CN1 en 3,3v. De même connecter la broche Tx du gps à la broche Tx du connecteur P1.

(la broche la broche IO35 du connecteur P3 est ainsi libre).

Attention, l'utilisation du Gps peut poser problème en mode débogage pour utiliser la liaison série usb de PlatformIO.

4 Connexion capteur de pression BMP180 ou MS5611 :

Il faut alimenter correctement le module par le connecteur CN1 en 3,3v. De même connecter la broche SDA à l'entrée 27 du connecteur CN1. Et la broche SCL à la pin 22 du connecteur P3 pour les ESP32-2432S028R ou à la broche NC/IO22 du connecteur CN1 pour les ESP32-2432S028.

5 Connexion capteur magnétique QMC5883 :

Il faut alimenter correctement le module par le connecteur CN1 en 3,3v. De même connecter la broche SDA à l'entrée 27 du connecteur CN1. Et la broche SCL à la pin 22 du connecteur P3 ou CN1. Capteur en parallèle sur le bus I2C avec le capteur de pression.

Ne pas oublier de faire la calibration logicielle du capteur magnétique pour son bon fonctionnement (voir documentation utilisateur). Pour le boîtier, je vais essayer de mettre le câblage et le haut parleur le plus loin de ce capteur.

6 Connexion de la batterie :

Connecter la batterie à la pin IO35 de la prise P3, par l'intermédiaire d'un potentiomètre de 10k ohms. Que l'on régler pour avoir la bonne tension affichée.

7 Alimentation :

Connecter la batterie 4,2v 1S lipo 4000 mah 965068 à la broche Vin de P1 par l'intermédiaire du convertisseur DC-DC 3,7v / 5V. Connecter la batterie au chargeur Usb C.

8 Réglage du potentiomètre :

Pour régler le potentiomètre, le but est d'avoir la bonne valeur affichée. Donc il faut mesurer la tension de la batterie (chargée de préférence à 4,2v) avec un voltmètre. Et ajuster le potentiomètre pour avoir la

bonne indication en page de boot ou en page Sys. Chaque ESP32 étant différent ce réglage est nécessaire et il faut garder à l'esprit que le convertisseur analogique digital n'est pas très linéaire, d'où la mesure de la tension haute de préférence. Ce réglage influe sur le barregraphe de % batterie en page Vz.

9 haut parleur :

Connecter le haut parleur à la prise P4 par l'intermédiaire d'une résistance de votre choix ou un shunt pour un très fort volume (haut parleur 2W 8 ohms + résistance 3,6 ohms dans mon cas) .

10 Utilisation du PCB :

Depuis la versions 20241010a un pcb est disponible. Connecter les différentes prises sur ce pcb. (prises JST 1.25)

11 Téléchargement du firmware :

Pour télécharger le firmware il y a 2 façons :

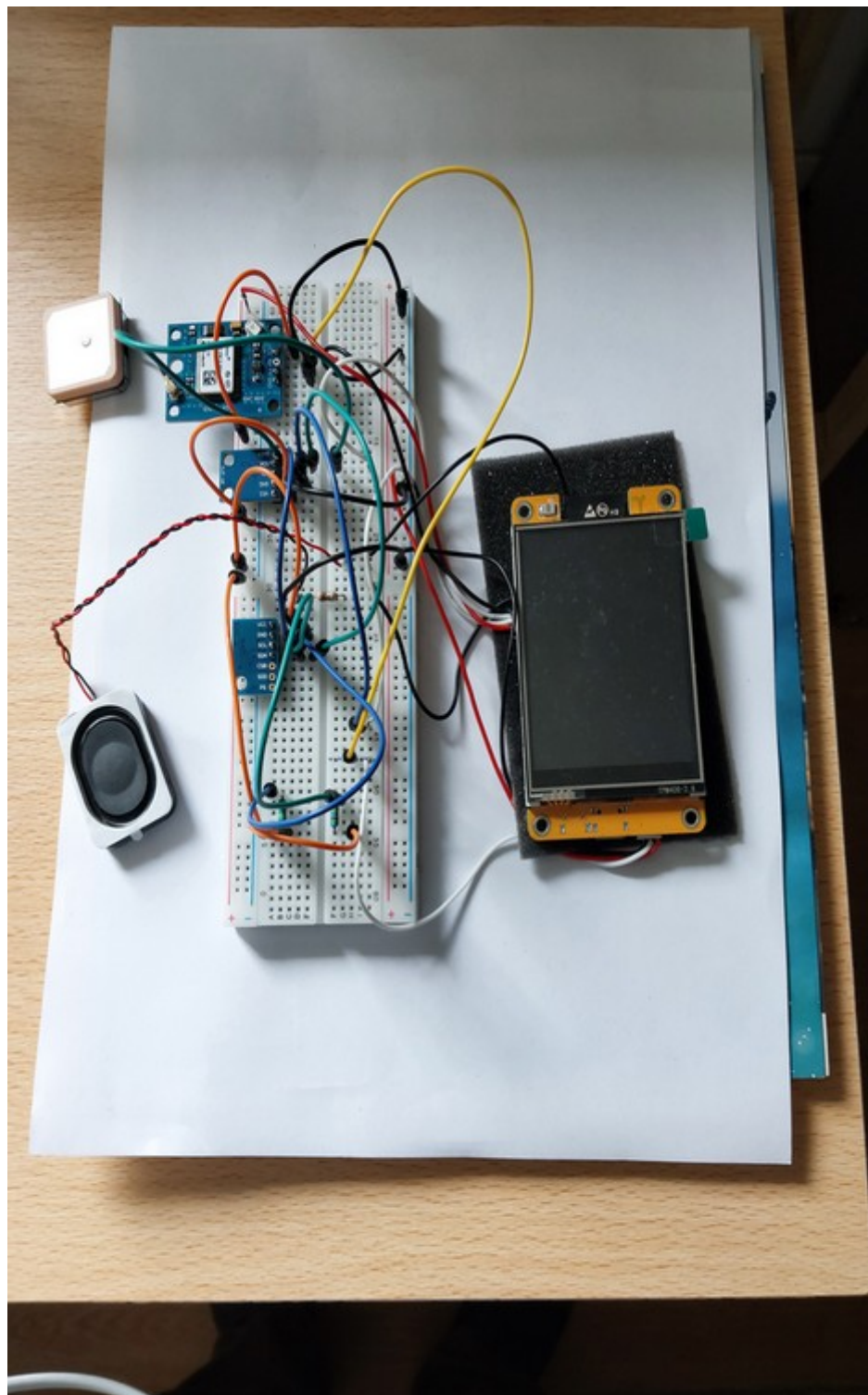
- soit par câble usb avec platformio ou un autre logiciel.
- soit par wifi. Il faut alors mettre le vario en mode téléchargement **Fir** en bas à droite de la page **Sys**. Et dé-commenter dans platformio.ini les lignes :

```
; extra_scripts = platformio_upload.py  
; upload_protocol = custom  
; upload_url = http://192.168.148.237
```

Avec bien sur l'adresse ip de l'url donnée par l'écran du BertheVarioTac. Cette procédure ne peut toutefois pas être faite au premier téléchargement de la carte ESP-2432S028 (R). Le logiciel ne s'y trouvant pas, il faut appliquer la procédure USB.

12 Quelques photos :

Le câblage sur plaquette sans alimentation avec capteur MS5611 :



Les premiers pas du boîtier fibre époxy :



Les composants de la partie alimentation (testés sur la carte seule) :

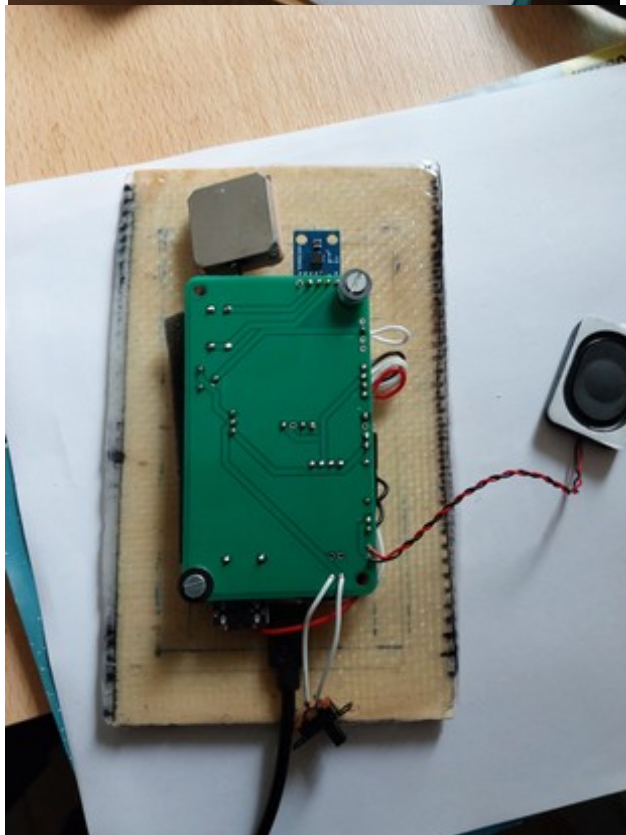
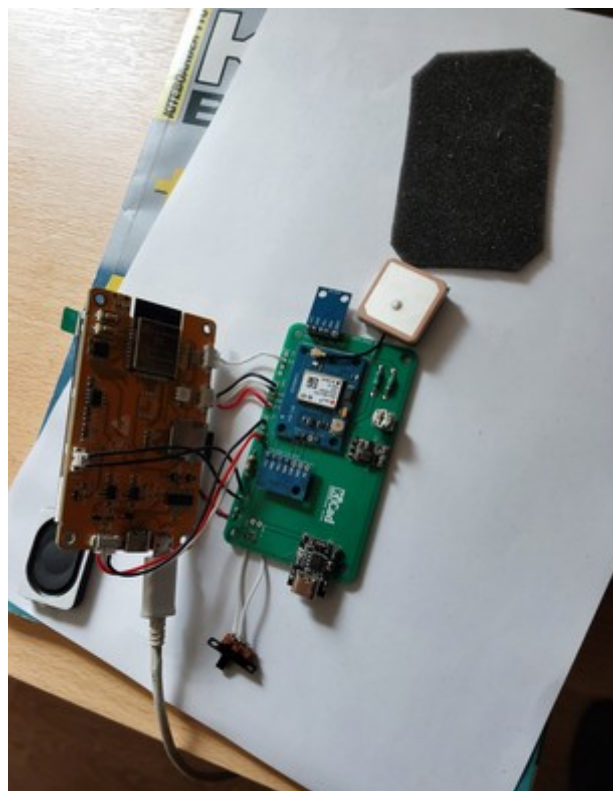
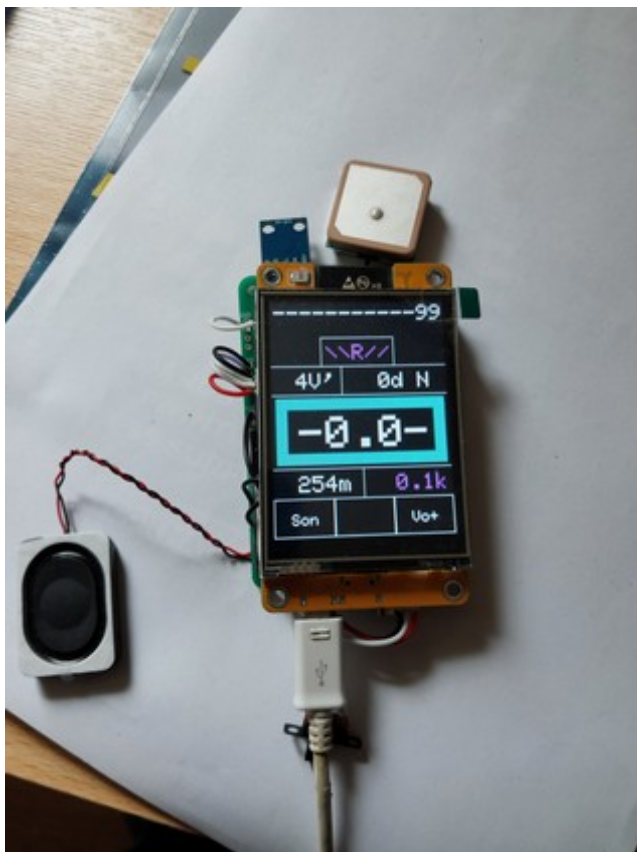




En attendant le PCB, poursuite du boîtier cagette/fibre/epoxy :



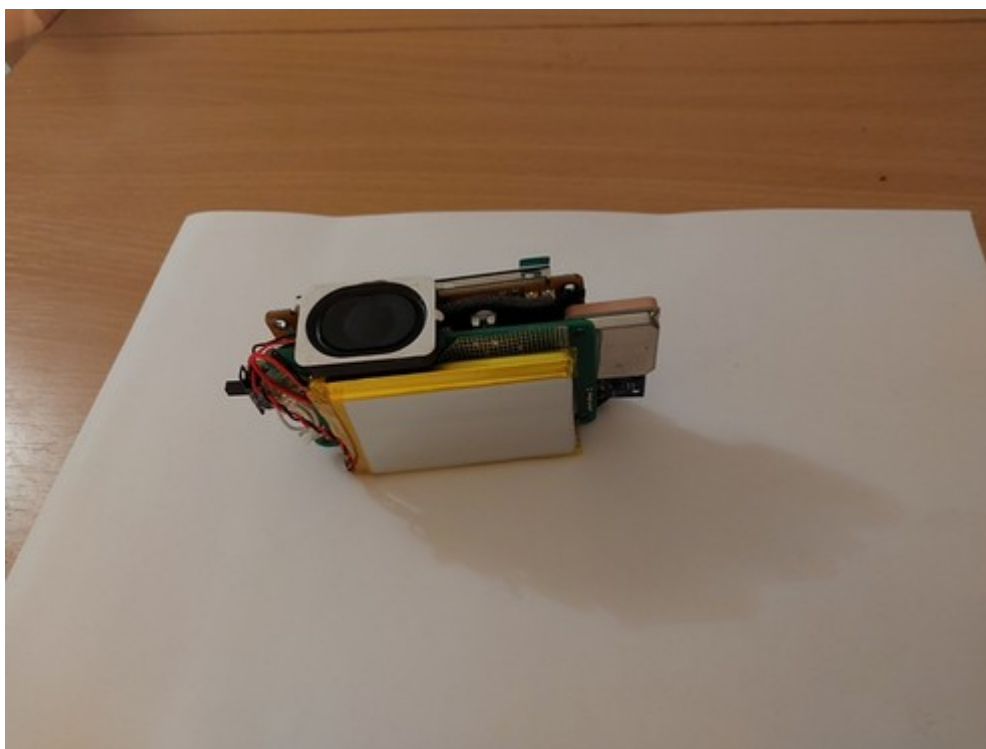
Le pcb fonctionne (sans la batterie pour l'instant) :



Le module électronique au complet en charge (fonctionnement sur batterie ok) :



finalement j'opte pour une batterie 4000mah modèle 965068. Autonomie 12h sans problème. Reste à terminer le boîtier.



Avec la fin de l'hiver je peut re-poncer de la fibre de verre et aérer. Poursuite du boîtier :



Phase finale du boîtier, reste plus qu'à coller le scratch et à faire les tests en vol.

