

Digibaro
Barographe Electronique

Introduction

Un nouveau concept de barographe en 2019?

Aujourd'hui, il est possible de siphonner en un instant les données de la station météo la plus proche sur son smartphone, ou d'obtenir les prévisions météo sur plusieurs jours pour n'importe quelle zone avec une précision incroyable. Cependant, si vous êtes un peu vieux jeu, décidez de naviguer dans une zone où la transmission de donnée est difficile, ou tout simplement êtes intéressé par les conditions locales, un baromètre reste un outil indispensable. Et un barographe, qui permet en un clin d'oeil d'évaluer la tendance, fournit un confort supplémentaire.

Les barographes mécaniques ne sont pas adaptés aux conditions sur un voilier. Il existe plusieurs barographes électroniques sur le marché, mais tous nécessitent d'être reliés à une source de courant externe.

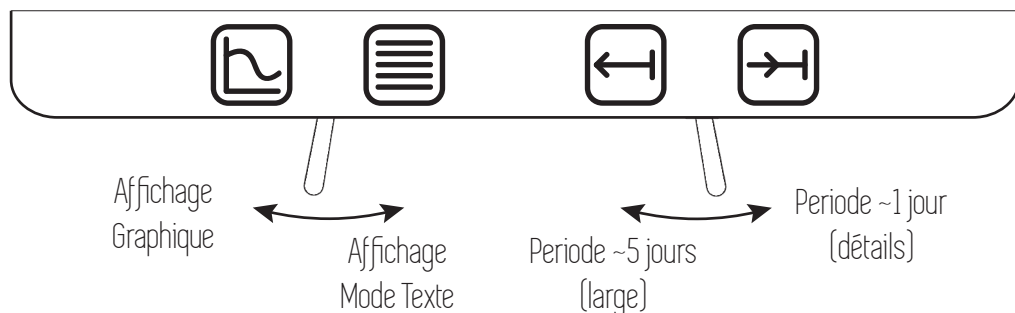
Le "Digibaro" est optimisé pour fonctionner de manière continue avec 3 petites piles rondes (AA) pour plus de six mois. De plus, son écran e-papier combiné avec une interface composée uniquement de 2 interrupteurs à bascule offre un design "analogique" unique d'un usage extrêmement simple.

L'ensemble du projet Digibaro est Open-Source, n'importe qui peut donc en construire une version et la modifier si besoin. Cela permet aussi de réparer facilement l'appareil puisque toutes les informations sont disponibles.

Voici le premier prototype du Digibaro, spécialement construit pour fêter les 80 ans de Papi Bateau!

Manuel d'utilisation

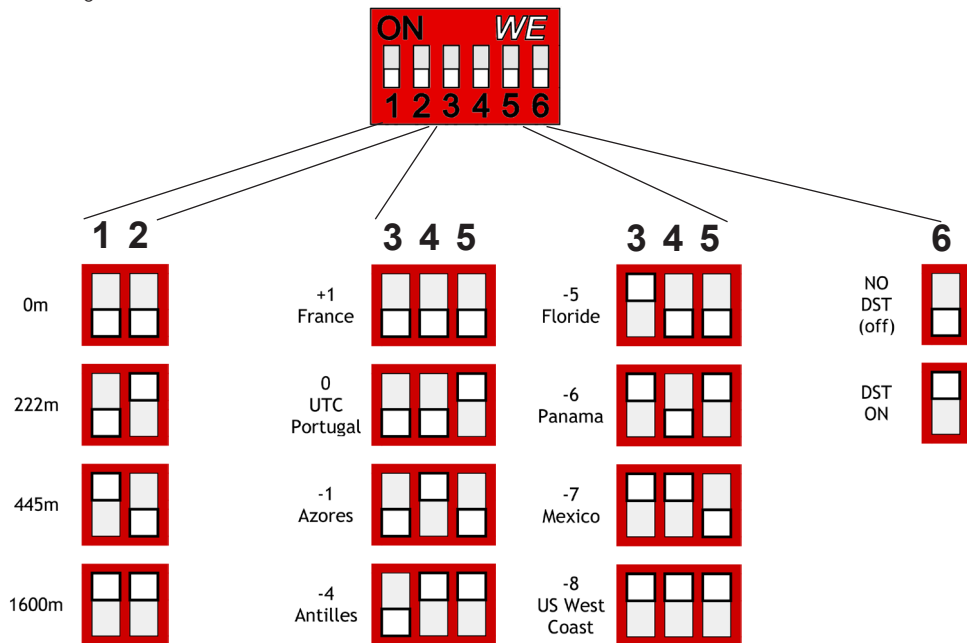
Deux interrupteurs à bascule permettent de sélectionner un des quatre modes d'affichage. Le Digibaro mesure la pression chaque minute, mais ne met à jour l'écran que toutes les 5 minutes. Le reste du temps, le Digibaro reste en sommeil profond, ne consommant que 0.12mW (environ 5000 fois moins qu'un smartphone avec écran éteint)! Chaque mouvement d'un interrupteur réveille le Digibaro et rafraîchit l'écran. Il faut environ 5 secondes pour rafraîchir ce type de e-papier, donc restez patient.



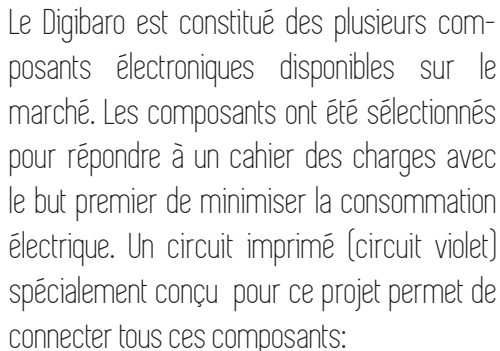
- » Interrupteur principal: accessible par une encoche en dessous du Digibaro, se trouve un interrupteur à glisser. Pousser vers le centre de l'appareil pour allumer, ou vers la gauche pour éteindre.
- » Changement des batteries principales: simplement enlever les trois piles AA (avec délicatesse, utiliser un outil pour aider à les sortir), et remplacer avec des nouvelles.
- » Pile de sauvegarde de l'horloge: tous les trois ans il faut changer la pile bouton. Suivre la procédure en dernière page afin de ne pas perdre l'heure.

Configuration des paramètres

La zone horaire ainsi que l'altitude sont configurés en utilisant les 6 boutons à glisser (DST=Day-light Saving Time=Heure d'été).

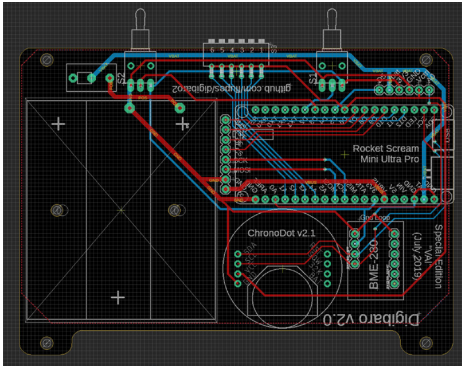


Il est fortement conseillé d'arrêter le Digibar en actionnant l'interrupteur principal (glissé vers la gauche=OFF) avant de voyager. Une fois sur place, configurez d'abord l'altitude appropriée, puis mettez en fonction le Digibar.



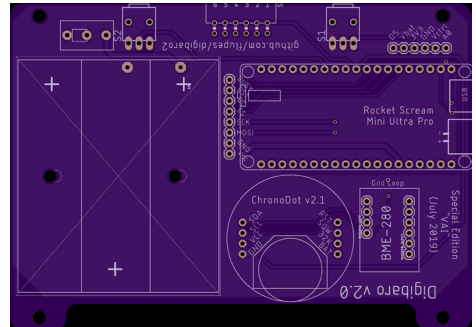
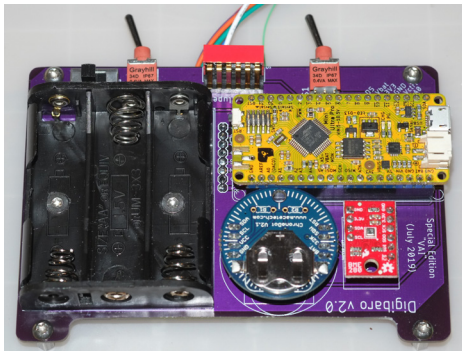
- » Le capteur de pression, température et humidité, le BME280 déjà placé sur une carte assemblée par Sparkfun (circuit rouge)
- » Une montre ultra-précise pour conserver le temps, la DS3231 montée sur la carte ChronoDot (circuit bleu).
- » Le microcontrôleur 32 bits reliant tous les périphériques et senseurs est basé sur un SAMD21 assemblé sur une carte de Rocket-Scream (circuit jaune)
- » L'écran "papier électronique" est assemblé

Du concept à la réalisation



par WaveShare. Il permet de conserver une image affichée alors que la source de courant est débranchée!

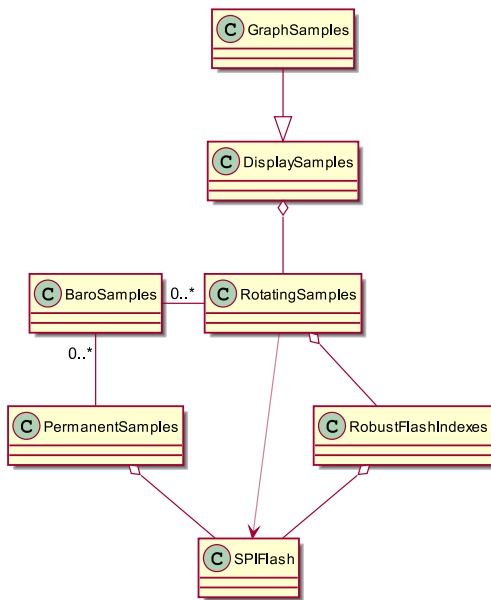
Les images sur cette page double présentent les étapes successives pour la création du système final.



Logiciel

Le logiciel pour Digibaro s'occupe de lire les informations des sensors et interrupteurs, de dessiner les graphiques et information textuelles. Chaque mesure météo est associée avec une date et heure précise, puis compressée afin de pouvoir stocker environ 25 ans de données horaires sur les 2MB (moins qu'une seule photo de smartphone) de mémoire permanente disponible!

Le système est conçu pour être résilient à toute interruption par un stockage fiable des informations détaillées des cinq derniers jours. Une attention toute particulière a été dédiée



pour permettre de configurer le microcontrôleur de manière à consommer de moins de courant possible en mode sommeil profond tout en restant à l'écoute du réveil de l'horloge ou des interrupteurs.

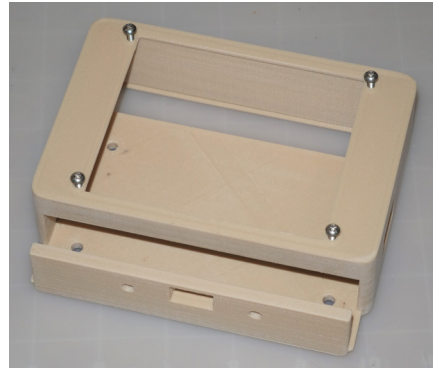
Code

```
423 void loop() {
424   static uint32_t after_awake = millis();
425   wdt_reset(); // Tell the watchdog timer everything is fine
426   loop_counter++;
427
428   // Enable power to the external RTC and display
429   digitalWrite(kRtcPowerPin, HIGH);
430   delay(50);
431
432   DateTime utc = ds3231_rtc.now();
433   SyncOnboardRTC(utc);
434   DateTime local = utc.getLocalTime(timezone);
435
436   // Get display mode from rocker switches state
437   uint8_t wake_switch_state = GetSwitchesState();
438   ConfigureSettingsFromDip();
439   BaroSample last_measurement = CollectSample(utc, local);
440   vbat_mv = MeasureVbat();
441
442   if (!timer_wakeup || utc.minute() % DAILY_PERIOD_MINUTE == 0) {
443     // Update display only to the period in minute defined above
444     // or if awakened by external switch input.
445     Display(local, last_measurement, wake_switch_state);
446   }
447
448   #ifndef KEEP_AWAKE
449   uint8_t new_switch_state = GetSwitchesState();
450   if (new_switch_state == wake_switch_state) {
451     // Only go in standby mode if switches were not modified
452     ConfigureForSleep();
453     awake_centiseconds += ((millis() - after_awake)) / 10;
454
455     // Go in standby mode for 1 minute!
456     // flash_debug.Message(FlashDebug::STANDBY, 1, loop_counter);
457     onboard_rtc.standbyMode();
458
459     // nNw we are awake again! --> detach the interrupts
460     onboard_rtc.detachInterrupt();
461     for (size_t p = 0; p < 2; p++) {
462       detachInterrupt(kSwitchesPin[p]);
463     }
464     after_awake = millis();
465     digitalWrite(LED_BUILTIN, HIGH);
466     // flash_debug.Message(FlashDebug::WAKEUP, 1, awake_centiseconds /
467     // (100*60));
468
469     CheckAndWaitForSerial();
470   }
471   #else
472   delay(10 * 1000);
473   #endif
474 }
```

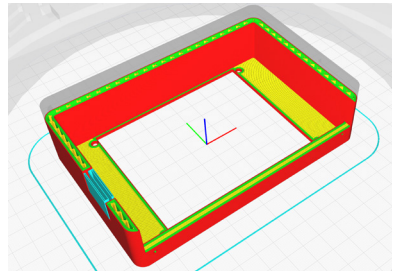
Probablement l'aspect du Digibaro qui a demandé le plus d'effort, le software nécessaire pour Digibaro, compte 5000 lignes de code (~100 fois l'extrait ci-joint). C'est donc un petit projet qui repose toutefois sur de nombreuses libraries totalisant environ 50'000 lignes de code. Par exemple, les routines de dessin bas-niveau sont développées par Adafruit. Ces libraries sont aussi Open-Source, ce qui permet, si nécessaire, de les adapter pour ce projet.

La plateforme logicielle Arduino, permettant de développer des applications pour le type de microcontrôleur utilisé, facilite grandement de type de projet.

The diagram shows a rectangular arena with a central platform. The arena is divided into four quadrants by a dashed line. The platform is a small square in the center. The arena is surrounded by a black border. The diagram includes labels for 'Platform', 'Arena', and 'Quadrant'.



A 3D perspective view of the experimental setup. It shows a rectangular container with a pinkish-red top edge and a yellow base. Inside, a large, dark, curved object (likely a turbine or propeller) is mounted on a central shaft. The shaft is supported by two bearings, each marked with a red crosshair. The entire setup is placed on a dark gray surface.



Spécifications

Précision barométrique absolue	± 1 mbar
Précision barométrique relative	± 0.1 mbar
Stabilité de l'horloge	± 2 ppm ~ 1 min / année max.
Résolution d'écran	400 x 300 pixels
Précision température	$\pm 1^{\circ}\text{C}$
Précision humidité	$\pm 3\%$
Consommation moyenne	> 6 mois avec 3 x AA (à valider ;-)

Changement de la pile de sauvegarde pour l'horloge:

1. Ouvrir le boîtier
2. Préparer la pile bouton de rechange
3. Eteindre l'appareil
4. Allumer l'appareil, et dès que l'écran affiche le message de bienvenue, retirer la pile bouton
5. Insérer la nouvelle pile moins de 20s après allumage

Open Source Hardware:

Les companies référencés dans cette brochure portent des noms sympas, mais en plus, elles contribuent activement à la communauté Open Source (hardware et software). Ceci au bénéfice des “makers”, et aussi de nombreuses autres organisations (comme la NASA). Elles méritent donc toutes une visite:

- » Adafruit <https://www.adafruit.com>
- » Arduino <https://www.arduino.cc>
- » ChronoDot http://macetech.com/store/index.php?main_page=product_info&products_id=8
- » RocketScream <https://www.rocket scream.com>
- » Sparkfun <https://www.sparkfun.com/>
- » WaveShare <https://www.waveshare.com>

Digibaro sur le web:

<https://github.com/flupes/digibaro2>

Digibaro a été conçu et assemblé en Californie par Lorenzo

Dessin icônes Digibaro et thème brochure par Kenzo