

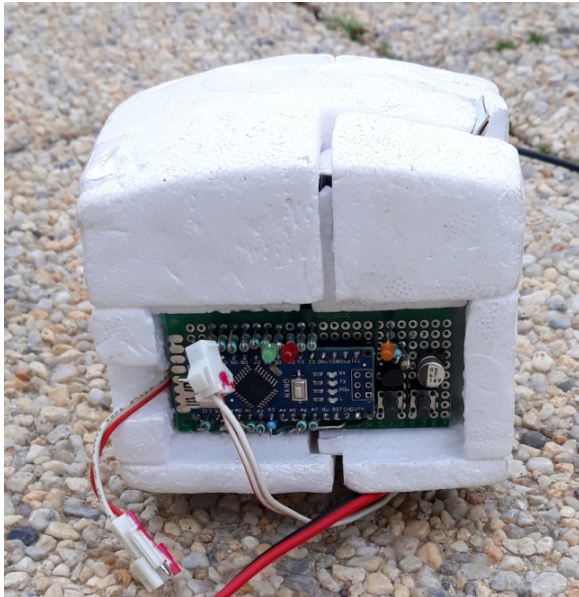
Ce montage réutilise un module ballon sonde météo **M10** pour en faire une **balise** d'exercice ou de 'manips', émettant des beeps réglables et un indicatif cw, sur des fréquences **uhf** hors météo. **Autonome** hormis pour configurer ou tracer. Il réutilise 'in situ' le pll **ADF7012** et le **GPS** du module 'M10'.

## CARACTERISQUES et particularités du montage

Balise **autonome** pour exercices et manips éducatives diverses.  
Emet des **beeps** audio réglables en fréquence, durée, séquence.  
Emet un **indicatif en CW**.  
Modulations : Sans, CW, **FM**, **AM** réglage du taux de modulation.  
Emet une **alarme** sur un **événement** déclencheur input ou analogique.  
**Horloge** soft autonome, synchro manuelle, PC et **GPS**.  
Installation, réglages et monitoring traces par **PC** via la liaison USB.  
Commandes et monitoring en **bluetooth** avec smartphone (facultatif).  
Commandes simples à 1 lettre par smart bouton en **CW**,  
**Modes** et fonctions : canaux préréglés, puissance, cycles, etc  
**Calibration** de la fréquence uhf.  
**Sortie série** GPS NMEA trames RMC GGA pour d'autres applis.  
Paramètres et configuration rangés indépendamment en **eeeprom**.  
**Alim** bloc 4 piles AA d'origine ou accus ou alim externe 5v ou 12v.

## COMPOSANTS principaux

Module ballon sonde météo **M10** [\*1] [\*2] 'tombé du ciel' [\*3]  
Circuit PLL **ADF7012** [\*4] bande UHF  
Circuit **GPS** Trimble-Copernik [\*5] [\*6] ou Sierra-FireFly [\*7]  
Antenne gps interne  
Arduino **NANO** [\*9] [\*10] petit et économique ou UNO mais plus grand...  
Module **bluetooth HC05** [\*13] (facultatif)  
1 bouton 1 'cw', 1 bt 2 'événement' ( ou prise), ( 3 bts présets ), prise 12v , fiches A/M, 3 leds Rouge, Verte et Jaune , (1 buzzer) , 1 fiche SMB antenne uhf.



Ph 1 le Nano, le connecteur de piles  
La languette du bt on/off



Ph 2 sortie SMB pour antenne

## Montage

Réalisé sur une plaque C.I. à trous portant le Nano, les composants du schéma et des liaisons par fils fins soudés par-dessous.

## Boîtier

Le **bloc polystyrène** suffit, avec des encoches pour 'coincer' l'arduino **Nano**, dans sa version de base autonome à 4 piles, la pile CR2032 et le HC05. Le brin d'**antenne** souple d'origine est réutilisé, via une fiche SMB mâle. Par temps humide le bloc sera mis dans un sac plastique !

## ON/OFF

Un connecteur 2 fils repérés noir et rouge, sert d'interrupteur entre le bloc pile et le M10. Le **bouton** on/off de la **languette** et sa led verte sont conservés.

'On' par appuie 5 secondes environ, la led verte clignote. 'Off' de même

## Liaison PC USB

Prévoir un câble modifié, fil rouge +5v pc 'open', ou un raccord M – F usb 5v open. Le PC **ne doit pas** alimenter le montage en 5v ou alors que le Nano **seul**.

## Liaison smartphone, tablette en USB Ph 8 (option )

Câble **mini-usb OTG** et soft interface RS232 usb, interface d'isolement du 5v

## Liaison smartphone, Bluetooth Ph 9 (option )

Module bluetooth HC05 [\*13] permet visu et ctrl 1 ou 2 lettres avec un Smartphone et 'Blueterm' pour les modes de bases, mais pas pour tous les réglages...

## Contrôle en local

Avec votre Rx sans l'antenne.

## Manips

tests d'antennes, petit 'géné', et chasse au renard ludique ou adrased pour simulation sonde 404Mhz, ou cospas 406Mhz avec un beep plus long...

## Quelques cas de réglages génériques :

**Beep** chaque seconde, le beep minute pleine est plus long suivi de l' indicatif

**Beep** toutes les x secondes **S** Séquence = x

**Beep** long minute et indicatif seuls **L** = x **S** Séquence = 0

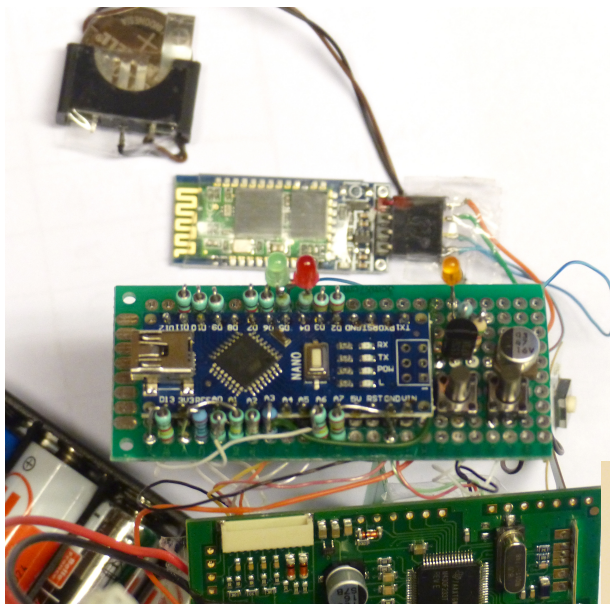
**Beep** long minute seul **L** = 2 **C** cw = 0 **S** Séquence = 0

**Beep** indicatif seul à heures pleines **O** schedule = 1

**Beep** indicatif sur événement ( si 'bt 2' ou une valeur prédéfinie en prg )

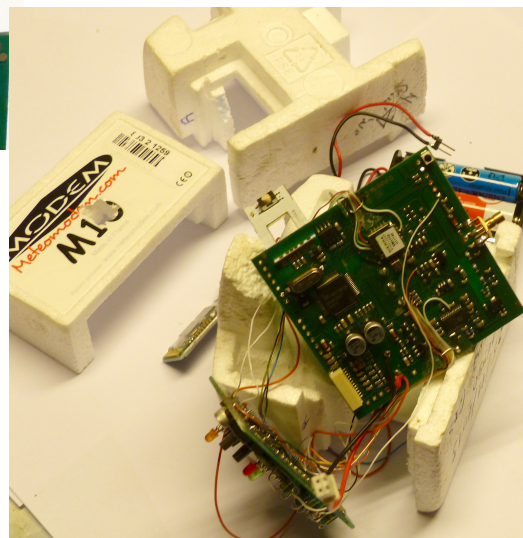
**Sortie** gps nmea en continu **y,l,s,c,o** = 0 **fct** XE

autres cas voir le prg.



Ph 3

La pile CR2032,  
Le bluetooth HC05  
Le Nano  
Le bloc à 4 piles  
Le M10



## Modes

En connexion PC USB avec l'IDE arduino/monitor série ou Hyperterminal, Hterm,  
Commandes par **PC** et/ou **bluetooth** à 1 lettre et extension 2 lettres X ...  
Commandes hors PC '**smart bouton**' en **CW** à 1 lettre , ( les 15 cas suivants )

**V** Veille commande **T** test comm , test l'état V ou B .  
**B** Balise mode normal de base

Les **8** modes suivants sont des **bascules oui/non** ou des **états tournants** des **n** cas possibles, avec mise à jour en eeprom si modifiés.

**P** power. 1 pll seul , env +10dBm, 2 avec le Tr driver du M10 env +23dBm  
**M** modulation, sans (porteuse), CW , **FM** , **AM** ( 0..3 )  
**Q** qrg choix **canaux** préenregistrées ( 1..4 ) ; Le (4) est réglable voir XJ  
**S** s beep **secondes** beep chaque n secondes ( 0 off ou 1..6) Synchronisé qtr gps  
**L** l beep **long minutes** (0 off ou 1..4) pour 1..20sec chaque Mn synchro qtr gps  
**C** c **indicateur CW** , 0 sans, 1 avec vitesse réglable voir define prg  
**O** Planification **horaire** cycle 24h : canal , veille , paramètres (schedule)  
**J** envoi indicatif CW

## Modes Presets

**1** Canal 1 Idem boutons bt preset (1 , 2 ,3 si installés)  
**2** Canal 2 4 fréquences par défaut  
**3** Canal 3 pré indiquées dans le source  
**4** Canal 4 canal 4 réglable , RIT + - , saisie et sauvegarde config eeprom

## Autres modes

**Y** y synchro 1 actif 0 veille totale  
**F** fixe 0 , mobile 1 option si GPS en mobile.  
**G** g contrôle affiche qtr gps UTC et locale  
**A** a alarme activée oui / non  
**N** n visualisation liste Modes **D** par défaut , **S** en système , et **Call** indicatif  
**H** help visualisation liste commandes, affichage, réglages  
**?** idem

**Fonctions étendues**      préfixe X ou x suivi de :

**Commandes , Contrôles, Réglages,**

- O** GPS init install (si type M10 Sierra)      auto. au démarrage
- D** GPS test 'raw' (brut) à vitesse PC usb
- G** GPS test 'raw' à vitesse nmea et Bluetooth sortie Serial pin 1 hors usb
- E** GPS test idem G mais en continue      fin par cr ou bt1
  
- M** mise à l'heure manuelle
- S** mise à l'heure synchro par le gps ( auto. à start + 3' et à 03H )
- C** entrée correction qrg calibration
- P** entrées paramètres note bf, durée beep, indice modul, indice power
- I** entrée indicatif du message cw , personnalisation
- J** entrée qrg ( en Hz ) par le **pc** pour le canal **4**
- Q** entrée grg ( en Hz ) par le **bluetooth** pour le canal **4**
- U** mode qtr UTC local 0, +1h, +2h      sera surchargé si synchro gps
- A** contrôle brut voies A/D
- T** contrôle des tensions (12v) et 5v et valeur approximative dBm antenne uhf
- K** contrôle envoie note fixe des minutes
- W** write met en eeprom config : call      note durée indices P et M
- R** reset valeurs par défaut du prg , idem si start avec bt1 cw appuyé
- +** fréquence + du pas préprogrammé ( 12,5kcs) (RIT) pour canal 4
- fréquence - du pas préprogrammé ( 12,5kcs)      //

**Contrôles et Tests en développement      X et**

- > ajustage puissance pll + entre 1 et 63      dyn environ 30db
- < ajustage puissance pll - entre 1 et 63
- { set pin modulation à 0      } set pin modulation à 1
- [ set indice power pll à 1 mini      ] set indice power pll à 63 maxi
- ( set indice modulation à 1 mini      ) set indice modulation à 63 maxi

**Action par le 'Bt smart' en cw**

Avec un appui sur le 'bt' smart, la led **verte** s'éteint, puis se rallume.  
Alors la saisie d'une lettre en cw est possible à une vitesse 'calme'.  
La led **rouge** suit la 'key'. Si pas de 'key' ou trop long ou lettre non permise  
erreur 4 pulses rouge... Autrement un pulse vert long termine la commande.  
Comme on est sans contrôle ! Une commande non souhaitée peut être faite !



### Etats Balise

La commande **N** affiche la liste des états Déf par défaut prg et Sys. en cours.

La commande '**XR**' remettra les valeurs par défaut .

Un démarrage avec le Bt 1 appuyé fait de même.

Le reset n'efface pas la personnalisation call, de la fct X I

### Etats LED

En balise la led **verte** 'pulse' chaque seconde, et la led **rouge** en émission.

En veille la led verte est **fixe**, retour auto en balise après 1 mn.

Le mode test état **T** relit l'état en cours V ou B et sert au test de dialogue PC.

Le cycle balise en cours peut être interrompu par le smart 'Bt 1' ou 'enter'.

En mode balise interrompue ou en veille les fonctions de saisies sont utilisables.

Une commande normale est terminée par un pulse led verte, autrement par 4 pulses led rouge ( option buzzer )

### Fonction Saisie heure manuelle XM

S'aider d'une horloge de bureau synchronisée DCF77 ou d'un PC à 'time calibré'.

Préparer la saisie, fonction '**XM**', date et l'heure au format indiqué avec une minute d'avance, puis au top 59 ... Valider lentement par 'enter'. Opt : PC direct

Contrôler avec la fonction '**T**' pour relire l'heure du cycle

### Mode Séquence S de beep des secondes

0 pas de beep des secondes, 1 à 5 beep chaque 1 à 5 secondes, 6 **aléatoire** 1..5

### Mode beep long L des minutes

0 pas de beep long, 1 beep idem seconde, 2 à 4 beep long note / durée en prg

Pour 5, 10 ou 20 secondes, l'indicatif cw, s'il est mis, sera alors à la seconde **30**.

### Fonctions GPS

Il doit être en extérieur ou à défaut près d'une fenêtre bien exposée.

'**XO**' réinitialise le gps de type Sierra

'**XD**' affiche les trames gps nmea pour monitoring pc .

'**XG**' fait de même en vitesse nmea et bluetooth

'**XE**' fait de même en continue , hors fiche usb, out pin 1

'**XS**' synchronise la qtr 'TIME' soft interne Nano, par le GPS.

La correction **UTC** du gps vers **H locale** sera déduite selon la **date**.

A la mise sous tension l'heure est par défaut 12h00 puis après **3 Mn**, temps théorique suffisant pour un '**fix**', une **synchronisation** automatique est lancée.

Le mode O schedule est prévu pour faire une synchro auto chaque 24h à 3h.

### Précision et stabilité en fréquence

La **précision** et la **stabilité** sont liées au petit quartz **8 Mhz** du uC M10 et à sa température. Il peut y avoir des petites différences entre M10.

### Réglage calibration XC

Exemple pour 400Mhz demandé, le signal est entendu à 399980 Khz soit 20Khz trop bas. Le pas est (  $8M / 32768$  ) de 244,14h. La correction 20k / pas donne ' 82'  
Valeur initialisée par défaut en prg ou en saisie XC pour le registre 0 et eeprom.

### Fonction saisie fréquence XJ ( PC ) ou XQ ( via bluetooth )

Saisie d'une fréquence uhf , saisie en **Hz**, pour le canal **4** . Le RIT est remis à 0

### Réglage fréquence RIT

**+** , **-** décalage de la valeur **12,5kcs** pré indiquée dans le prg ( canal 4 seul )  
NB à la précision près du pas du pll  $8Mhz / 2^{15}$  244hz

### Mode choix canaux Q

Choix tournant du canal 1..4. Fréquences QRG pré indiquées dans le source soft

### Fonction XP saisie paramètres

Choix de la **note bf**, de sa **durée**, de l'indice de **modulation** de l'indice de **puissance**. La note des minutes est pré réglée dans le source du prg.  
Les valeurs saisies sont 'limitées' à 400..2300hz, 100..900mS, 1..63, 1..63  
NB les indices de pwr et modul sont **cumulatifs**, en AM la somme doit être < à 63

### Fonction XI saisie indicatif

Choix d'un indicatif pour le **message cw** , qui est aussi utilisé comme alarme  
**Personnalisation** avec un Nano déjà préprogrammé avec entrée d'un call f5xyz  
Par **XI** puis avec mise en eeprom **XW**

### Mode choix power P

Choix d'une puissance **P1** pll seul env. +10 dBm , ou **P2** avec le driver env. +23 dBm ( 200mW ) si valeurs indice mod, indice pwr au max, et état pin modulateur.  
Impact, selon l'exercice souhaité, sur l'autonomie des piles ...

**Fonctions réglages pwr** et indice de modulation ( en AM ) ou déviation ( en FM )  
>, < pas à pas , [ , ] , { , } , ( , ) , **préfixés** . Manips avec M modul. 0 : porteuse

### Mode alarme A

Une action 'bt 2' ou une tension capteur (A1) déclenche une alarme par l'émission de l'indicatif, si l'action est activée ou pas par **A** et/ou les autres fonctions et modes à 0.

NB une photorésistance et si variation brusque de lumière ou un capteur I.R.

### **Fonction XR Reset**

Remet les valeurs par défaut indiquées en prg. Un start 'on' avec le bt 1 cw appuyé fait de même

### **Fonction XW Write**

Range en eeprom les configs modifiées : call, fréquence du canal 4, note , durée ; indice modul, indice power, correction ( en auto après saisie : option voir prg )

### **Fonction XO planificateur horaire ou scheduler**

Par défaut envoi l'indicatif en cw chaque heure, dans les modes courants.

D'autres valeurs de capteurs peuvent être envoyées en cw ...

Ce mode permet de reprogrammer, par tranches horaires ( pleines )

tous les paramètres balise, fréquence, note, mode, etc

Une commande de re - synchro qtr gps est lancée à 3h

Une fonction sommeil (option) permet de désactiver le pll et le gps du M10

et invalide les leds, afin de réduire au minimum la consommation

La commande réveil réactive ces fonctions.

### **Etats monitoring modes et commandes** à 1 lettre ou à 1 chiffre ou X extension.

Avec le mode 'outils' 'monitor série' de l' IDE arduino ou Hyperterminal.

Les états successifs sont affichés majoritairement avec un N° de 'trace' #nn

Les états en cours sont affichés 1 fois quand une commande est donnée ou T

### **La séquence** de base indique la QTR et les secondes en progression.

Suivies de **B** ou **L** aux beeps des Sec et/ou des Mn si mis.

La fonction **O** limite les traces aux minutes et aux tests horaires.

### **Présets canaux** avec les 'Boutons 1 , 2 , 3' ( non câblé, option )

Commandes directes pour changement de canal, les paramètres associés sont prédéfinis dans le source. L'état en cours en eeprom n'est pas modifié.

**Mode fixe ou mobile F** Cette option utilise les ressources du GPS et de la bibliothèque TinyGPS++ pour location,vitesse, azimuth, tracking destination etc



## Limitations et options possibles

L'application permet une utilisation d'un PLL ADF7012 externe , [\*4]  
une sortie NMEA ttl ( si modèle Sierra ), une sortie pulse 1 PPS.

Le Nano est à son maximum en entrées, sorties, mémoires PRG et RAM !

Un autre ARDUINO ou un TEENSY 3.2 [\*12] plus conséquent en mémoire et entrées/sorties et avec un chip RTC externe (ds3231) ou interne , permettraient des modes M mobile et une modulation plus numérique. ( cf f5zeh )

Modulation NRZ ( APRS), FT8 etc, avec les possibilités gfsk et ook du adf7012

Revoir les #define dans balm10beep.h en rapport.

Evolutions possibles location degrés décimaux, locator, tracking distance azimuth vers ou depuis un point cible, transmission position en aprs ( cf f1zbx-3 )

Balise avec ajout de capteurs météo, ou autres ( cf R3, Multifonctions)

Enregistreur de tracés, nécessite un add-on carte micro-SD formaté FAT.

NB WSPR UHF 432,300Mhz ! La commande data devra modifier un peu la fréquence du xtal 8Mhz du uC avec une varicap. Comme il faut décaler en fsk 4 paliers espacés de 1,46hz et avec le rapport de 50,5 la méthode sera très délicate...

## Interface 5v <--> 3,3v

Le Nano est en **5v** et le M10 en **3,3v** ou 3v pour les pll et gps et le uC.

Il est spécifique d'interfacer via des petits modules quadruples à **FET** .

La méthode simple employée ici, avec des **résistances de 2,2k**, si elle n'est pas 'académique', fonctionne. Les R 'limitent' les courants de pull-up over tension et inversement ne diminuent pas les 'level high' , les courants étant très faibles.

Le Nano normalement s'accommodant de 3v en entrée '1'.

## GPS

Il doit fournir en **4800** (ou 9600bds) une trame **NMEA** avec **RMC** et **GGA** [\*5]

La sortie 5v du nano vers l'entrée 3v peut-être via une 2,2 Kohms

Module M10 avec un Trimble Lassen Copernicus **4800 Bds**

Le module doit être reprogrammé préalablement avec TrimbleMon [\*6]

la sortie est « ttl » 3v vers le 5v d'un Nano mais suffit via une protection de 1K.

Module M10 avec un **Sierra** Ph 4 FireFly  
[\*7] **38400 Bds**

Le module doit être reparamétré à chaque démarrage ( en situation V 0.9 )

par des commandes du protocole **MTK**. [\*14]

Option 'flash' en étude [\*8]

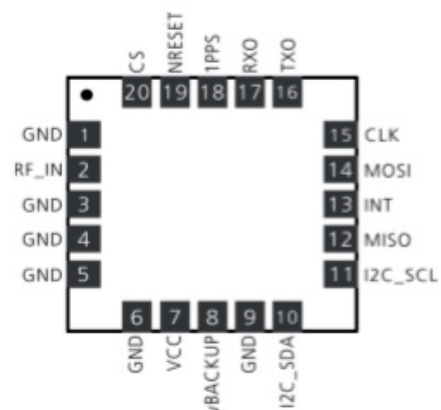


Fig. 2.3 Pin Configuration

## Trames

Les 'trames' exploitées sont **GGA** et **RMC** et suffisent pour la date et l'heure .  
Les éléments de position x,y,z sont aussi exploitables , cf biblio TinyGPS++  
Notons que la réception série, le traitement et le cycle diffèrent d'une seconde le temps réel reçu.

En mode 'raw' XE les trames GGA RMC sont aiguillées en **sortie Serial** pin 1 TX pour utilisation par un soft externe. Le connecteur USB étant alors débranché.  
Paramètre XD pour vitesse pc usb et XE pour vitesse gps 4800

## GPS 1PPS

Ces modules fournissent un signal ' 1PPS ' **un pulse par seconde**, qui est un top 0 de référence de temps de la seconde. Actif quand le 'fix' de suffisamment de satellites est acquis.

Le signal active une led de visualisation de ce ' fix ' via un transistor NPN tampon.  
Il peut aussi entrer facultativement sur une pin en interruption pour des ' manips ' nécessitant plus de précision. Ou pour d'autres manips : GPS-DO 10Mhz [\*15]

## M10

Le module M10 émet de base des infos météo particulières, humidité, température, pression mais implique un 'déplacement' qui recalcule celle-ci en fonction des différences de positions x,y et d'altitude z. [\*1] à M10

Il ne faut évidemment pas émettre sur les canaux météo 400 à 406 Mhz.

La M10 serait préprogrammée sur 403Mhz et observée sur 404Mhz.

D'autres types de ballons utilisent 409,910Mhz.

Le 432Mhz peut ne pas être possible. Voir adaptations soft ou de la micro self du pll !

La modulation est FSK 4800bds. Une trame de 100ms par seconde à 200mW, une autonomie de près de 30 heures.

Manip : Sur un module d'origine à 404Mhz, un quartz de 8,588Mhz de récupération permet de générer une porteuse à 433,694 Mhz ! ( coeff. 50,5 )

## PREREQUIS

Document pdf **ADF7012** [\*4] ; manips Bluetooth avec **HC05** ou **HC06**

**Sur M10** Interface mini-usb – USB pour dév. et RS, interface usb 5v coupé.

## PROGRAMMES Annexes

**Sur PC** Arduino IDE, Hyperterminal, Hterm, GPSviewer 1.8

## Librairies

Arduino EEprom SoftwareSerial Teensyduino [\*12]  
Github : JChristensen , Mikalhart Tinygps++  
pjrc : Paul Stoffregen , Margolis time timelib AltSoftSerial  
pecan de KT5TK liens voir le prg  
Sur smartphone Serial USB Terminal avec cordon mini-usb/ OTG/USB pour  
Interface série usb – RS , Bluetooth Terminal, Blue Term [\*16]  
Autres références f5bez : quelques partiels réutilisés...  
MORSE ( ass. Intel 8080 Tx Rx 1979 f1bez ) , R3 V2 et RU3 V1 en 8085,  
R3 V4 Teensy2++ , MULTIFONCTIONS Teensy 3.2, AMORSE 2011 ( Teensy 2 )  
F5ZEH balise 28Mhz 2014 , BALWSPR 2019

## Anomalies

Alims. Fonctions Y, L, S à 0 ? ou O à 1 ? faire Mode 'XR' reset défauts.  
Vérifier liaisons , antennes, le signal 8Mhz du uC  
Vérifier la QRG au fréquencemètre. PII hors gamme de rattrapage...

**Consommation** selon manip et pwr entre 90 mA et 170 mA ( en 5v )

**Manips** Accus et mini panneau solaire... et fonction sommeil ( avec led off )

**Précautions** tensions parasites et statiques : les modules uC, PLL, GPS, et Nano sont fragiles ! Fer à souder bien isolé voire débranché lors des soudures aux pins...

## Options boîtier

Boîtier alu et alim 'propre' bloc 220v à 12v, ou 12v véhicule  
Un régulateur 7805 pour le 5v.

## Affichage LCD

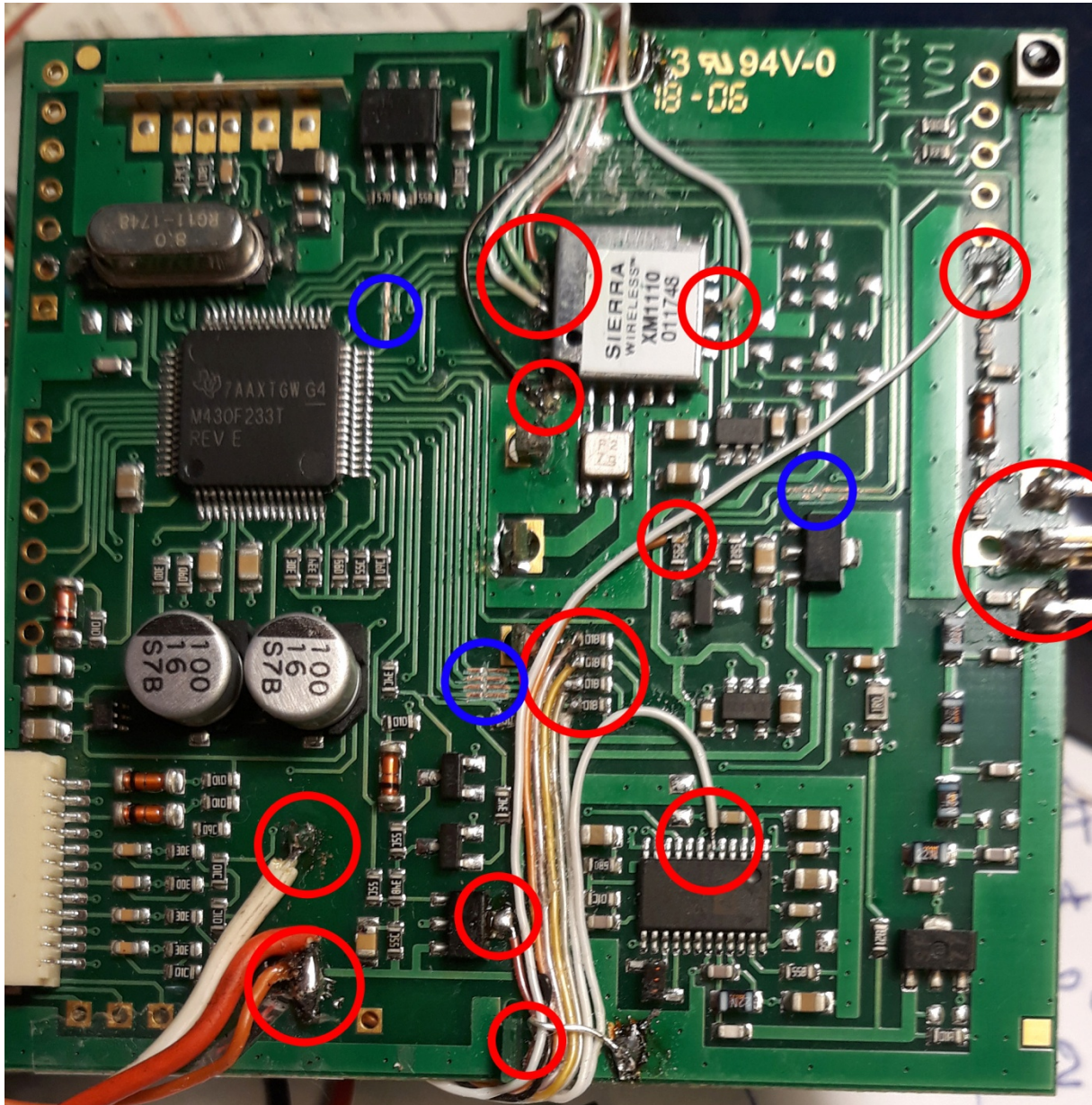
Ce montage en Nano, n'utilise pas de LCD. La finalisation comme 'box' tx balise autonome permanente 'sans visu' suffit .

## Histoire de bips ou beeps...

Le 4/10/1957 le 1<sup>er</sup> satellite 'Spoutnik' faisait la même chose ! [\*19]

## REALISATION

Le module M10 est 'scarifié' de 6 coupures sur le C.I. au scalpel. ( en bleu ). 14 petits fils fins monobrins pré étamés diamètre 0,22mm 0,46mm sur isolant, pris sur un ex câble réseau, ( en rouge ) , sont ajoutés et attachés sur le bord du M10 par un pont soudé. Prévoir assez long de fils, pour l'installation ou pour un UNO externe. Il faut une panne très fine, une bonne loupe et pas de « palu breton » ce jour là...

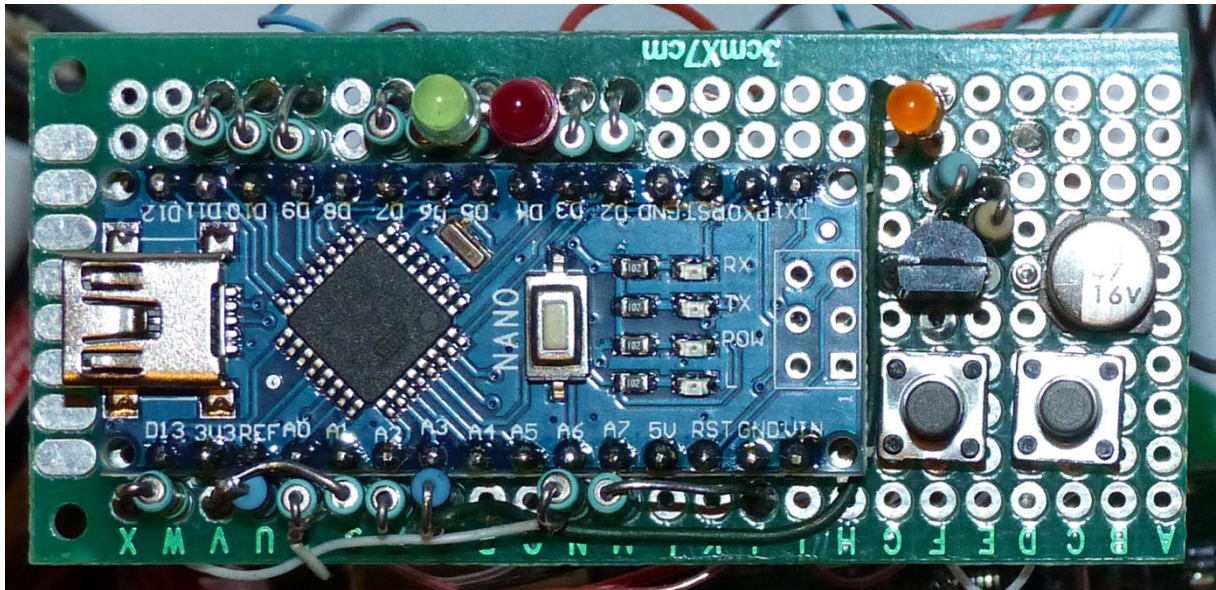


Ph 5 Les 'modifs' du M10 « M10+ V01 »

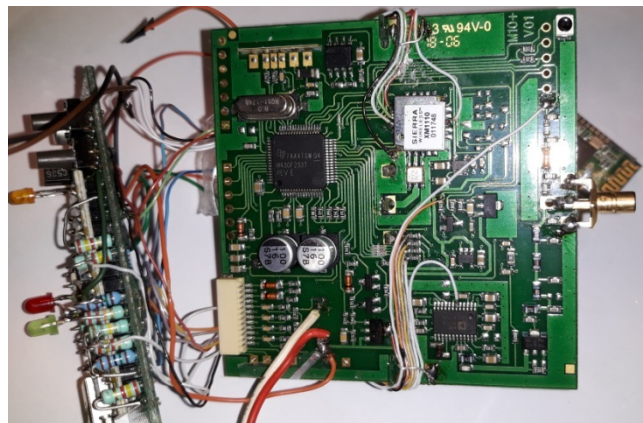
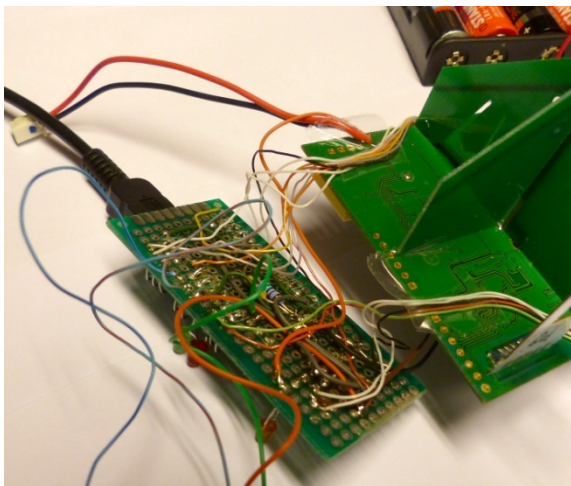


## NANO

Le Nano est installé sur une plaque C.I. époxy à trous. La plupart des pins sont munies d'une résistance verticale de 2,2kilohms. Les quelques composants boutons et leds sont câblés par des petits fils en dessous. Au final, les fils ajoutés du M10 sont reliés au Nano. S'aider du schéma. Ph 5,6,11



ph 6 Le Nano, les R tampons, les leds , le Tr led jaune, le bt 1 cw , le bt 2 alarme



Ph 7,8 Les liaisons du M10 sont soudées aux résistances tampons des pins

## Comm



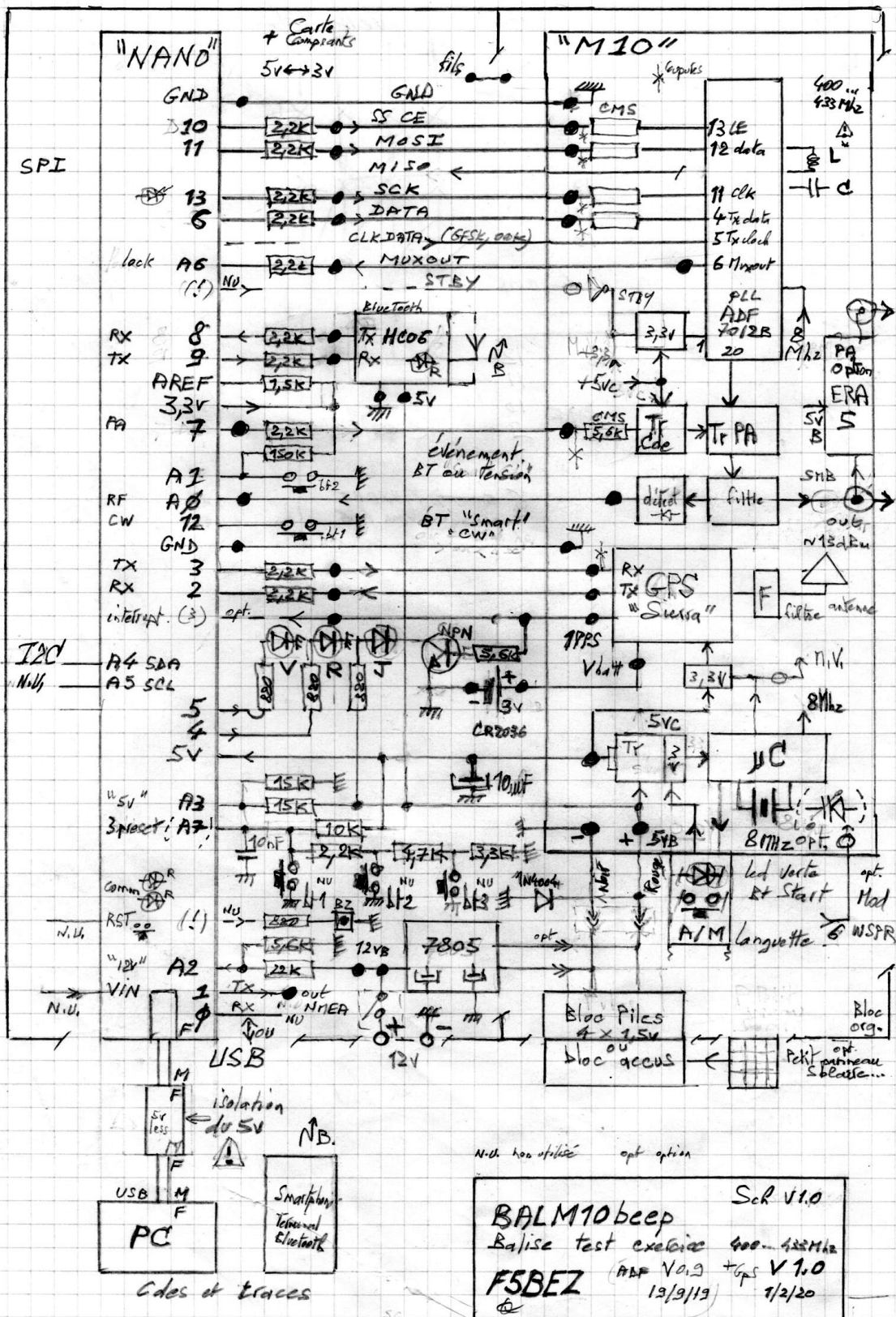
Ph 9 Smartphone avec  
l'appli  
' **Serial USB Terminal** ',  
le câble mini-usb **OTG**,  
le séparateur de 5v  
et la bague ferrite.

Ph 10 Smartphone avec  
l'appli

' **Bluetooth Terminal** '







## PROGRAMME

L'étude des installations sous arduino, des ajouts de bibliothèques, du 'C' arduino est conseillé en s'aidant des nombreux sites éducatifs à cette fin...

### Mini guide d'installation

Installez d'abord l'outil interface de développement **arduino...** ( IDE ) [\*11]

Testez votre **Nano** neuf **seul**, avec un câble USB 'normal' c.a.d. avec le 5v.

La 1<sup>ère</sup> connexion du NANO sera affichée en bas à droite du PC, un port com sera créé. Le Nano exécute probablement un programme 'blink' ( clignotant ) préchargé.

Lancez l' IDE Arduino puis indiquez

Outils/Type de carte/Arduino Nano

Outils/Processeur/Atmega328P (Old Bootloader) **OU** Atmega328P selon les nanos

Outils/Port/com4 ( 4 par exemple )

Fichier/Préférences > Préférences

Afficher les résultats détaillés pendant : compilation

Cochez 'afficher les numéros de lignes' 'vérifier le code après téléversement'

Prenez le programme standard test 'clignotant'.

Fichier/Exemples/Basics/Blink précisez led interne port 13

Sous 'fichier' clic icône 'vérifier', puis la flèche 'téléverser'. Prêt pour la suite !

Maintenant il vous faut ajoutez des '**bibliothèques**' 'Teensyduino' [\*12]

Des indications complémentaires et des liens sont indiqués dans le source

Depuis l'IDE Outils/Gérer les bibliothèques > gestionnaire de bibliothèque saisie **Time** (M Margolis) .. 'installer'

Une bibli déjà installée est grisée et est marquée 'INSTALLED'

Téléchargez **TinyGPSPlusMaster**, liens dans le source, GitHub, Arduiniana

Déplacez le zip (qui a un sous-dossier src) ou le dossier vers votre dossier

balM10beep. Croquis/inclure une bibliothèque/Ajouter la bibliothèque zip/ la bibli

Vous verrez ensuite ces bibliothèques à 'Contributeur bibliothèques'

Elles seront placées en [votrecompte]\Documents\Arduino\librairies ( dont Time )  
c:\Programmes(x86)\Arduino\hardware\teensy\avr\librairies

pour des choix de cartes de la famille Teensy. Ces biblis sont réutilisables...

Dézipper balM10beep.zip dans votre dossier **BALM10beep**

Fichier/ouvrir/balM10beep.**ino** . Le programme est structuré par catégories, les onglets montrent les **.cpp** et **.h** associés. La version avec un Nano contraint à une optimisation des variables : les textes fixes sont en zone mémoire programme.

Des explications complémentaires sont indiquées dans les sources.

Lire d'abord balM10beep.**h** Vous pouvez alors 'vérifier' puis 'téléverser' .

Les noms des fonctions, macros, variables et commentaires sont explicites au

mieux ! Utiliser la fonction 'trouver', 'chercher dans tous les onglets' pour pister

Voir certains '#define' pour des fonctions non utilisées.

AVRDUDE permet de charger un .hex précompilé, voir sites spécialisés

## Mise en œuvre

Vous avez déjà une expérience arduino, alors vous repartez des sources.

Vous adaptez à vos préférences : call, fréquences..

Vous demandez à un ami OM de vous programmer un Nano avec votre call.

Vous avez un Nano déjà programmé, vous le personnaliser avec la fonction XI.

Cette description, se veut vulgarisatrice et explicative. Le logiciel est libre pour le domaine radioamateur. Il ne peut en être fait d'usage commercial de tout ou partie. Les noms et indicatifs des auteurs dans les librairies et modules doivent être indiqués.

Dominique LEVEQUE **F5BEZ** 2020

## Documentations

Différents points, réalisations et exemples

<http://www.meteofrance.fr/prevoir-le-temps/observer-le-temps/moyens/les-radiosondages>

[\*1] <http://www.meteomodem.com> « Meteomodem »

<http://www.meteomodem.com/m10.html>

[\*2] <http://radiosonde.eu/RS01/RS01A.html#01>

<http://radiosonde.eu/RS03/RS03H02.html>

[\*3] Merci aux « chasseurs » du « 35 » Maurice, F4EQD, F4SGU, F4GNZ, etc

[\*4] le PLL ADF7012

<https://www.analog.com/media/en/technical-documentation/data-sheets/ADF7012.pdf>

[\*5] [https://fr.wikipedia.org/wiki/NMEA\\_0183](https://fr.wikipedia.org/wiki/NMEA_0183) GPS Protocole NMEA

GPS NMEA autonome à partir d'une sonde M10 version Trimble

[\*6] [http://ara35.fr/wp-content/uploads/2018/01/BALLON-SONDE-METEO-M10-V3-modifs-v2\\_parF5BEZ.pdf](http://ara35.fr/wp-content/uploads/2018/01/BALLON-SONDE-METEO-M10-V3-modifs-v2_parF5BEZ.pdf)

[\*7] <https://source.sierrawireless.com> airprime gps modules GPS Firefly

[\*8] Manips en cours F4EQD , cf documentations FireFly

[\*9] <https://store.arduino.cc/arduino-nano> officiel

[\*10] [https://www.ebay.fr/b/Nano-arduino/bn\\_7006324116](https://www.ebay.fr/b/Nano-arduino/bn_7006324116) alternative ch.

[\*11] <https://www.arduino.cc/en/Main/Software> outil développement V1.8.12

[\*12] [https://www.pjrc.com/teensy/td\\_download.html](https://www.pjrc.com/teensy/td_download.html) biblios V1.51

[\*13] [https://www.ebay.fr/sch/i.html?\\_sacat=173036&\\_nkw=hc-05+arduino](https://www.ebay.fr/sch/i.html?_sacat=173036&_nkw=hc-05+arduino)

[\*14] protocole PMTK

[https://www.sparkfun.com/datasheets/GPS/Modules/PMTK\\_Protocol.pdf](https://www.sparkfun.com/datasheets/GPS/Modules/PMTK_Protocol.pdf)

[https://cdn-shop.adafruit.com/datasheets/PMTK\\_A11.pdf](https://cdn-shop.adafruit.com/datasheets/PMTK_A11.pdf)

[\*15] <http://ara35.fr/wp-content/uploads/2019/02/GPSDO-F5BEZ18.pdf>

[\*16] <https://play.google.com/store/apps?hl=fr> rechercher

[\*17] <https://www.pjrc.com/teensy/>

[\*19] <https://fr.wikipedia.org/wiki/Spoutnik>

[https://fr.wikipedia.org/wiki/Fichier:Sputnik\\_beep.ogg](https://fr.wikipedia.org/wiki/Fichier:Sputnik_beep.ogg) et le « son » beep !

## Traces

//// Démarrage qtr base 12 :00 date D 01/01/2020

//// Cycle beep chaque seconde indicatif à seconde 1

#30:PowPA :1

05B 06B 07B 08B 09B

H 12:00:10 B 11B 12B 13B 14B 15B 16B 17B 18B 19B

////

H 12:00:50 B 51B 52B 53B 54B 55B 56B 57B 58B 59

D 01/01/2020

B

H 12:01:00 B 01

#33 Ind.CW: F5BEZ

B 05B 06B 07B 08B 09B

H 12:01:10 B 11B 12B 13B 14B 15B 16B 17B 18B 19B

////

H 12:01:50 B 51B 52B 53B 54B 55B 56B 57B 58B 59

D 01/01/2020

B

H 12:02:00 B 01

#33 Ind.CW: F5BEZ

B 05B 06B 07B 08B 09B

H 12:02:10 B 11B 12B 13B 14B 15B 16B 17B 18B 19B

////

H 12:02:50 B 51B 52B 53B 54B 55B 56B 57B 58B 59

D 01/01/2020 <<<<

B

///// à qtr start + 3mn synchro gps

H 12:03:00 GPS>QTR

#41:Essai:1

#47:GPS ??

#50:Flux?

#52:GPS data 1

#53:GPS vu

#48:N2S:5

UTC 14:13:34 05/02/2020 4 Mercredi

#49:TZ +1

QTR>TIM

LOC 15:13:34 05/02/2020 4 Mercredi

#42:GPS:93 TIM:0

#43:GPS ok

Fin

B 35B 36B 37B 38B 39B

H 15:13:40 B 41B 42B 43B 44B 45B 46B 47B 48B 49B

H 15:13:50 B 51B 52B 53B 54B 55B 56B 57B 58B 59

D 05/02/2020 <<<<<<

B

H 15:14:00 B 01

///// heure date et beeps synchros par le gps

//// Démarrage avec le bluetooth non connecté au smartphone

InitBT OK ok est la réponse du module HC05

DoneBT

BALM10beep f5bez V1.0

SetParDef

//// Démarrage avec le bluetooth connecté au smartphone

InitBT Connect le module ne renvoie pas ok s'il est déjà connecté

DoneBT

BALM10beep f5bez V1.0

SetParDef

/// chgt note  
H 17:46:20 B 21B 22B 23B 24B 25B 26B 27B 28B 29B  
H 17:46:30 B 31B 32B 33B 34B 35B 36B 37B  
RxPc:P  
Action:57  
DebFct:57  
note duree indam indpwr  
1000,500,50,30  
23009003030  
fait  
Fin Test:0

RxPc:

Action:0  
51#21:Synch:1  
#22:Canal:1  
#23:Freq:400000000  
#24:Mode:fm  
#25:Periode:1  
#26>Note:2300  
#27:Duree:900  
#28:indmod:30  
#29:indpwr:30  
#30:PowPA:1  
52B 53B 54B 55B 56B 57B 58B

//// tests 2 modes Pwr  
#29:Indpwr:31  
#30:PowPA:2 <<<<  
#31:  
pll:255 hf: 0.27 V  
57B#31:  
pll:255 hf: 0.29 V +20 <<<<

RxPc:P  
Action:12  
Pwr choix;1  
Write EEp  
59

#21:Synch:1  
#22:Canal:1  
#23:Freq:400000000  
#24:Mode:no  
#25:Periode:1  
#26>Note:1000  
#27:Duree:300  
#28:Indmod:31 <<<< à 1/2  
#29:Indpwr:31 <<<< à 1/2  
#30:PowPA:1 <<<<  
#31:  
pll:255 hf: 0.25 V  
H 19:10:00 B#31:  
pll:255 hf: 0.04 V +3 <<<<

01B#31:  
pll:255 hf: 0.04 V

///  
#21:Synch:1  
#22:Canal:1  
#23:Freq:400000000  
#24:Mode:no <<<<< porteuse

