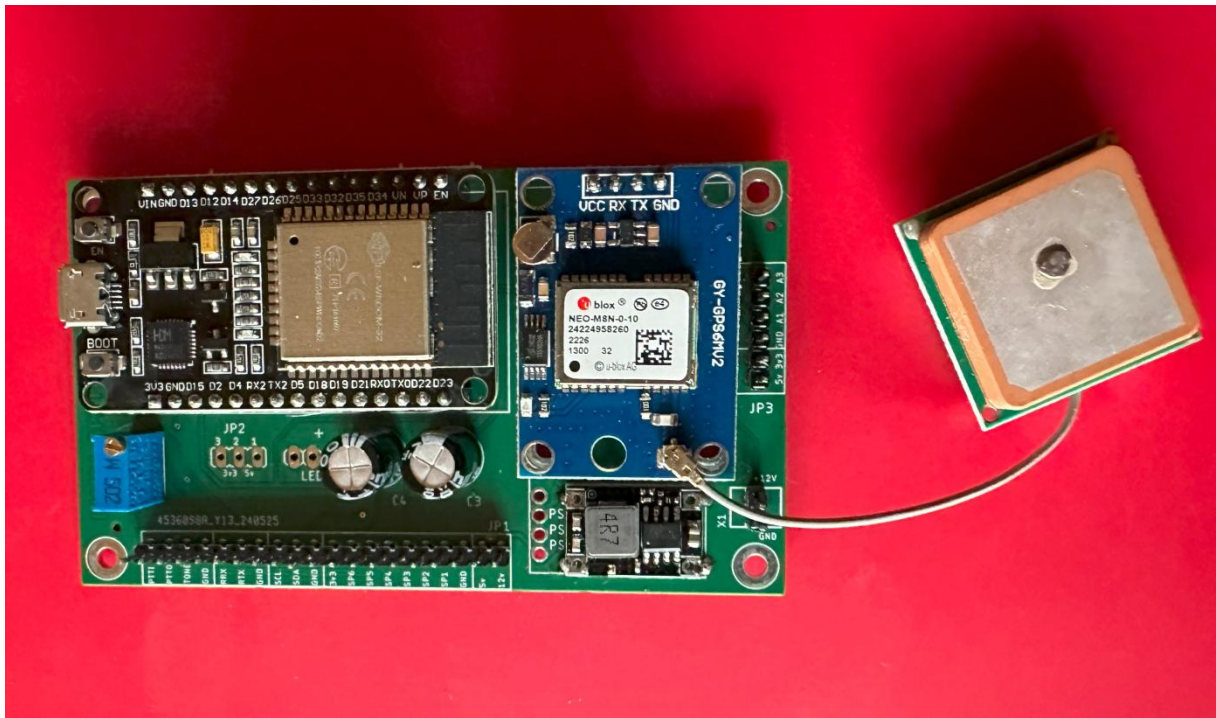


FbBalise – Balise CW

FbBalise est un gestionnaire de balise basé sur un module microcontrôleur ESP32.



Il permet de générer des séquences programmables par script. Une interface WEB permet de configurer le configurateur.

L'accès à la configuration se fait par un ordinateur ou smartphone en se connectant au réseau Wifi « FbBalise » avec le mot de passe « 12345678 » par défaut. Le nom du point d'accès et son mot de passe sont modifiables par configuration.

Caractéristiques principales :

- Interface de configuration Wifi
- Alimentation 7-18V (30 à 70mA suivant configuration)

Sommaire

1. Démarrage	3
2. Page accueil.....	4
3. Page Configuration.....	5
3.1. Paramètres WiFi	5
3.2. Option tempo	5
3.3. Sortie du signal OOK télégraphie.....	5
3.4. Inversion signal OOK télégraphie.....	5
4. Réalisation du script.....	6
4.1. Création du script.....	6
4.2. Script simple	6
4.3. Script plus évolué.....	6
5. Mots clé du script	7
5.1. DEBUG	7
5.2. DELAY	7
5.3. IF / ELSE / ENDIF.....	7
5.4. LOOP / ENDLOOP	8
5.5. MODE	8
5.6. SEND.....	8
5.7. SET.....	8
5.8. TUNE.....	9
6. Variables prédéfinies.....	10
7. Expressions.....	11
8. Comparaisons	12
9. Remerciements	13

1. Démarrage

FbBalise est configuré par des pages WEB accessibles par Wifi. A la mise sous tension, quelques secondes sont nécessaires pour que le Wifi soit activé.

Depuis votre ordinateur personnel ou votre smartphone, cherchez le réseau Wifi « FbBalise » et connectez-le.

Après la connexion au Wifi « FbBalise », entrez le mot de passe si nécessaire (« 12345678 » par défaut), une fenêtre d'accueil s'affiche au bout de quelques secondes. Cette fenêtre va permettre d'accéder par un menu aux différentes pages de l'application.

Si vous êtes bien connecté au Wifi « FbBalise » mais la fenêtre ne s'affiche pas, vous pouvez la connecter à l'adresse <http://192.168.4.1>

2. Page accueil

La page d'accueil propose un accès à :

- La configuration par la roue dentée en haut à droite de la fenêtre,
- Quatre onglets permettant de gérer et exécuter quatre scripts différents.
- Une fenêtre texte éditable qui permet de créer ou modifier le script, avec la position du curseur sous la fenêtre.
- Une ligne affichant un éventuel message d'erreur ou une ligne de debug.
- Un bouton permettant de démarrer, redémarrer ou sauvegarder un script suivant le contexte
- Une ligne de status donnant l'état d'exécution du script.

3. Page Configuration

Cette page permet de modifier la configuration par défaut. La sauvegarde de la configuration est faite par l'appui sur le bouton sauvegarder.

3.1. Paramètres WiFi

- Nom du réseau WiFi (SSID)
- Mot de passe pour l'accès à ce réseau WiFi.

3.2. Option tempo

- Le Wifi s'active à la mise sous tension du PWS.
- Si « Tempo » est actif, le Wifi s'arrêtera automatiquement au bout du temps spécifié en minutes.
- Le temps « Tempo » minimal est de 5 minutes. Toute action de changement de page ou de sauvegarde remet à zéro le compteur de temps.
- Si l'option est validée, le point d'accès WiFi ne sera accessible qu'après coupure et remise sous tension.

3.3. Sortie du signal OOK télégraphie

- Sélection du port de sortie CW Key (collecteur ouvert ou spare 1 à 6)

3.4. Inversion signal OOK télégraphie

- Inverse le signal de sortie CW

Toute modification devra être enregistrée par le bouton « Valider »

3 boutons en bas de fenêtre vous permettent de choisir un type de redémarrage :

- « Redémarrer » : le FbBalise est redémarré, les paramètres sont conservés, aucune modification n'est faite au PWS.
- « Effacer la flash » : les scripts contenus dans la mémoire flash sont effacés . Le FbBalise redémarre et la configuration est conservée.
- « Paramètres usine » : tous les paramètres et scripts sont effacés, le FbBalise redémarre en configuration par défaut.

4. Réalisation du script

4.1. Création du script

Le script est entré au clavier (ou collé) dans la fenêtre d'édition. Lorsque l'édition est terminée, le bouton de sauvegarde va envoyer et sauvegarder le script dans la flash de l'ESP32.

- Le script sera interprété en majuscules et ne fait pas de différences entre majuscules et minuscules.
- Toute ligne devra commencer par un mot clé, suivi de ses arguments éventuels.
- Toute ligne vide est ignorée.
- Toute fin de ligne sera ignorée après le caractère #

4.2. Script simple

Ce script va permettre de :

- Sélectionner le mode CW à 15 mots/minute, son à 800Hz
- Envoyer l'indicatif et le qth locator suivi d'une pause de 1s, tune de 5s, pause de 1s
- Le tout en boucle infinie.

```
MODE CW 15 800
LOOP
    SEND "F6FBB JN12BT "
    DELAY 1000
    TUNE 5000
    DELAY 1000
ENDLOOP
```

4.3. Script plus évolué

Ce script va générer :

- Un mode CW 15 mots minute à 800Hz,
- trois fois l'indicatif, le qth locator et une pause de 500ms,
- ensuite, pause de 500ms puis tune de 5 secondes puis pause de 1 seconde
- il bouclera indéfiniment...

```
MODE CW 15 800
LOOP
    LOOP 3
        SEND "F6FBB JN12BT "
        DELAY 500
    ENDLOOP
    DELAY 500
    TUNE 5000
    DELAY 1000
ENDLOOP
```

5. Mots clé du script

- Un paramètre entre crochets est optionnel
- Une expression peut être
 - une valeur numérique
 - une chaîne de caractères entre guillemets
 - une expression entre parenthèses, contenant des expressions, variables, valeurs numériques, parenthèses et signes +, -, /, *, %
 - une condition entre parenthèses, contenant des opérations, variables, valeurs numériques, parenthèses et signes >, >=, <, <=, =

5.1. DEBUG

- **DEBUG** a un paramètre texte affiché sur la page d'accueil, sous la fenêtre d'édition.
- Utiliser des guillemets pour définir l'argument car il peut contenir des espaces.
- Si l'argument contient une variable, son contenu sera transformé et affiché en texte.

```
DEBUG "TENSION: $($_A2 * 5.12)VOLTS UPTIME : $_UPTIME"
```

- Dans cet exemple, on suppose un pont diviseur externe d'un rapport de 5.12.

5.2. DELAY

- **DELAY** a un paramètre expression spécifiant le délai de pause en millisecondes

```
DELAY (EXPRESSION)
```

5.3. IF / ELSE / ENDIF

- **IF** a un paramètre entre parenthèses définissant la condition.
 - Si la condition est différente de 0, seule la partie IF sera exécutée.
 - Si la condition est égale à 0, seule la partie ELSE sera exécutée.
- Un **ENDIF** doit obligatoirement spécifier la fin du test conditionnel.

```
IF (CONDITION)
```

```
...
```

```
ELSE
```

```
...
```

```
ENDIF
```

5.4. LOOP / ENDLOOP

- LOOP peut avoir un paramètre optionnel spécifiant le nombre de boucles
- Si le nombre de boucles n'est pas spécifié, la boucle sera infinie
- Un ENDLOOP doit obligatoirement spécifier la fin de boucle.

LOOP [(EXPRESSION)]

...

ENDLOOP

5.5. MODE

- Mode attend un ou plusieurs paramètres spécifiant les options du mode
 - CW (vitesse en mots / minute) [tonalité]
 - OPERA PERIOD [tonalité]
 - PERIOD parmi OPERA05, OPERA1, OPERA2, OPERA4, OPERA8, OPERA32, OPERA65, OPERA2H

MODE CW (EXPRESSION) [(EXPRESSION)]

MODE OPERA OPERA05 [(EXPRESSION)]

5.6. SEND

- SEND permet d'envoyer un texte passé en argument en utilisant le mode précédemment défini par le mot-clé MODE.
- Utiliser des guillemets pour définir l'argument car il peut contenir des espaces.
- Si l'argument contient une variable, son contenu sera transformé en texte
- Dans le cas de OPERA, seul in indicatif valide sera accepté en argument.

SEND "F6FBB JN12BT "

SEND "TENSION \$(\$_A2 * 4.1) " # diviseur de tension externe par 4,1

5.7. SET

- SET permet de définir ou redéfinir la valeur d'une variable.
- Une variable doit commencer par le caractère \$.
- Une variable est exclusivement numérique.
- Une variable peut être utilisée dans une expression.
- SET a deux paramètres :
 - Un paramètre spécifiant le nom de la variable. Ce nom doit obligatoirement commencer par une consonne et être exclusivement alphanumérique.
 - un paramètre expression spécifiant la valeur numérique affecté à la variable

SET \$NOMDEVAR (\$AUTREVAR / 12.5)

SET \$NOMDEVAR 12.5

SET \$NOMDEVAR (\$NOMDEVAR + 1) # incrémente \$NOMDEVAR

5.8. TUNE

- TUNE a un paramètre expression spécifiant le temps du signal en millisecondes

TUNE (EXPRESSION)

6. Variables prédéfinies

Plusieurs variables relatives au système sont prédéfinies. Il s'agit toujours de variables numériques. Leur nom commence toujours par un underscore (tiret bas) excepté pour \$(expression)

Variable	Valeur	Commentaire
\$(EXPRESSION)	Expression incluse	Contient l'évaluation de l'expression incluse
\$_I1 A \$_I6	0 ou 1	Etat du spare en entrée 0=LOW 1=HIGH
\$_O1 A \$_O6	0 ou <> 0	Positionne le spare 0=LOW sinon HIGH
\$_A1 A \$_A3	0 à 3,3	Tension en entrée sur l'analog en volts
\$_HOUR	0 à 23 ou -1	Donnera -1 si le GPS n'est pas actif
\$_MINUTE	0 à 59 ou -1	Donnera -1 si le GPS n'est pas actif
\$_SECOND	0 à 59 ou -1	Donnera -1 si le GPS n'est pas actif
\$_DAY	1 à 31 ou -1	Donnera -1 si le GPS n'est pas actif
\$_MONTH	1 à 12 ou -1	Donnera -1 si le GPS n'est pas actif
\$_YEAR	yyyy ou -1	Donnera -1 si le GPS n'est pas actif
\$_UPTIME	Nombre de jours	Nombre de jours depuis le démarrage
\$_MILLIS	millisecondes	Millisecondes depuis le démarrage du script
\$_LOOP	Nombre de boucles	Nombre d'itérations dans la boucle active



Les variables \$_I1 à \$_I6 et \$_A1 à \$_A3 positionnent le port spare en entrée. Les variables \$_O1 à \$_O6 positionnent le port spare en sortie. Si le port spare concerné est utilisé pour un autre usage, cela peut affecter son fonctionnement.



La tension en entrée sur un port spare ne doit pas dépasser 3,3v sous peine de détruire le microcontrôleur. Les mesures de tension supérieures peuvent se faire en utilisant un pont diviseur externe. La valeur peut alors être multipliée en utilisant une expression.

Exemples :

```
SET $_O2 1 # Positionne le port spare 2 au niveau haut
```

```
SET $_O2 0 # Positionne le port spare 2 au niveau bas
```

```
IF ($_I1) # Vrai si le port spare 1 est à niveau haut
```

```
IF ($_A3 > 2.1) # Vrai si le port analog 3 est à une tension supérieure à 2,1v
```

7. Expressions

Une expression est de la forme « A opérateur B » où :

- A est une valeur, une variable ou une expression.
- Un opérateur du tableau ci-dessous,
- B est une valeur, une variable ou une expression.

. Les opérations sont réalisées en virgule flottante.

Les opérateurs sont :

Opérateur	Action	Commentaire
+	Addition	
-	Soustraction	
*	Multiplication	
/	Division	Quotient de la division flottante
%	Modulo	Reste de la division flottante

Les multiplications, divisions et modulo sont effectuées en priorité, de gauche à droite.

Les additions et soustractions sont effectuées en deuxième priorité, également de gauche à droite.

Une expression peut être entourée de parenthèses, auquel cas les expressions entre parenthèses sont effectuées en priorité.

Exemples :

$(3 + 5 * 2)$ $\Rightarrow 3 + 10$ $\Rightarrow 13$

$((3 + 5) * 2)$ $\Rightarrow 8 * 2$ $\Rightarrow 16$

$(3 * 5 + 2)$ $\Rightarrow 15 + 2$ $\Rightarrow 17$

8. Comparaisons

Une comparaison est de la forme « A Comparateur B » où :

- A est une valeur, une variable ou une expression.
- B est une valeur, une variable ou une expression.

Les calculs sont faits en virgule flottante. L'égalité peut donc être difficile à trouver. Le résultat d'une comparaison est égal à 1.0 si le résultat est vrai sinon 0.0 si le résultat est faux.

Les opérateurs sont :

Comparateur	Action	Commentaire
<	Inférieur	
>	Supérieur	
<=	Inférieur	
>=	Supérieur ou égal	
!= ou <>	Différent	
== ou =	Egalité	

Exemples :

IF (\$_A3 > 2.1) # Vrai si le port analog 3 est à une tension supérieure à 2,1v

Exécute si analog3 supérieur à 2.1v

ELSE

Exécute si le le port analog 3 est à une tension inférieure ou égale à à 2,1v

ENDIF

9. Remerciements

La réalisation logicielle est de ma conception. Ce logiciel fonctionne sur une carte FbBeacon et pourra utiliser une carte fille GPS (pour la mise à l'heure) **(A faire...)**

Le logiciel est écrit en C/C++ sous l'environnement « platformio / vscode » et inclut plusieurs librairies tierces ESP32 .

Merci à F5DJL et F1HDI pour leurs idées, remarques et soutien !

Jean-Paul ROUBELAT.