



COMPTE-RENDU D'EXPERIENCES



BALLON G2L - CLUB NODAL'S

Rédigé par Olivier Sterenberg



SOMMAIRE

1. Présentation du club.....	3
2. Présentation du projet.....	3
2.1. Les objectifs.....	3
2.2. La conception et la réalisation.....	4
2.2.1. Carte d'alimentation.....	5
2.2.2. Emetteur KIWI Me et modulation FSK.....	5
2.2.3. Capteurs de température interne et externe.....	6
2.2.4. Capteur de pression.....	6
2.2.5. Système de positionnement.....	7
2.2.6. Système de prise de vues.....	7
3. Suivi du ballon.....	7
3.1. Le logiciel de réception.....	7
3.2. La chasse au ballon.....	8
4. Résultats de la télémessure.....	10
5. Conclusion.....	11
6. Remerciements.....	11
7. Annexes.....	12



1. Présentation du club

Nodal'S a été créé en septembre 2004 par deux élèves ingénieurs désireux de mettre en œuvre un projet technique et scientifique, en guise d'application directe de toutes les connaissances acquises en cours. La démarche expérimentale, au cœur des activités du secteur Espace de Planète Sciences, a permis au club de mener à terme la réalisation d'une nacelle de ballon stratosphérique dans le cadre des activités Espace mises en œuvre par le CNES et l'Association.

2. Présentation du projet

2.1. Les objectifs

G2L est le premier projet scientifique mis au point par Nodal'S. Il consiste en la réalisation d'une nacelle dont les objectifs principaux sont de :

- ✓ Ecrire un soft de réception de télémesure permettant d'afficher :
 - les données mesurées par les différents capteurs embarqués,
 - les positions du ballon et de la voiture obtenues grâce, respectivement, à un GPS embarqué et un GPS à main,

et d'enregistrer toutes ces données dans des fichiers binaires et csv,

- ✓ Ecrire un soft embarqué à destination de l'émetteur KIWI Me, seule CPU de la nacelle,
- ✓ Développer une technique performante de chasse au ballon permettant de retrouver facilement la nacelle (technique différente de la méthode de repérage par triangulation des radioamateurs).

Pour la réalisation de ces objectifs, nous nous sommes appuyés sur la mise en œuvre d'expériences relativement simples comme :

- ✓ La mesure des températures interne et externe à la nacelle,
- ✓ La mesure de la pression dans les différentes couches de l'atmosphère,
- ✓ La prise de vues de la rotondité de la Terre,
- ✓ Le positionnement du ballon par GPS.

L'originalité de ce projet réside dans le fait que la seule CPU est celle du KIWI Me qui a été reprogrammé pour l'occasion.



2.2. La conception et la réalisation

Le schéma synoptique suivant (cf. Figure 1) décrit l'interconnexion entre les différents blocs constituant le projet. Chacun de ces blocs est décrit ci-après.

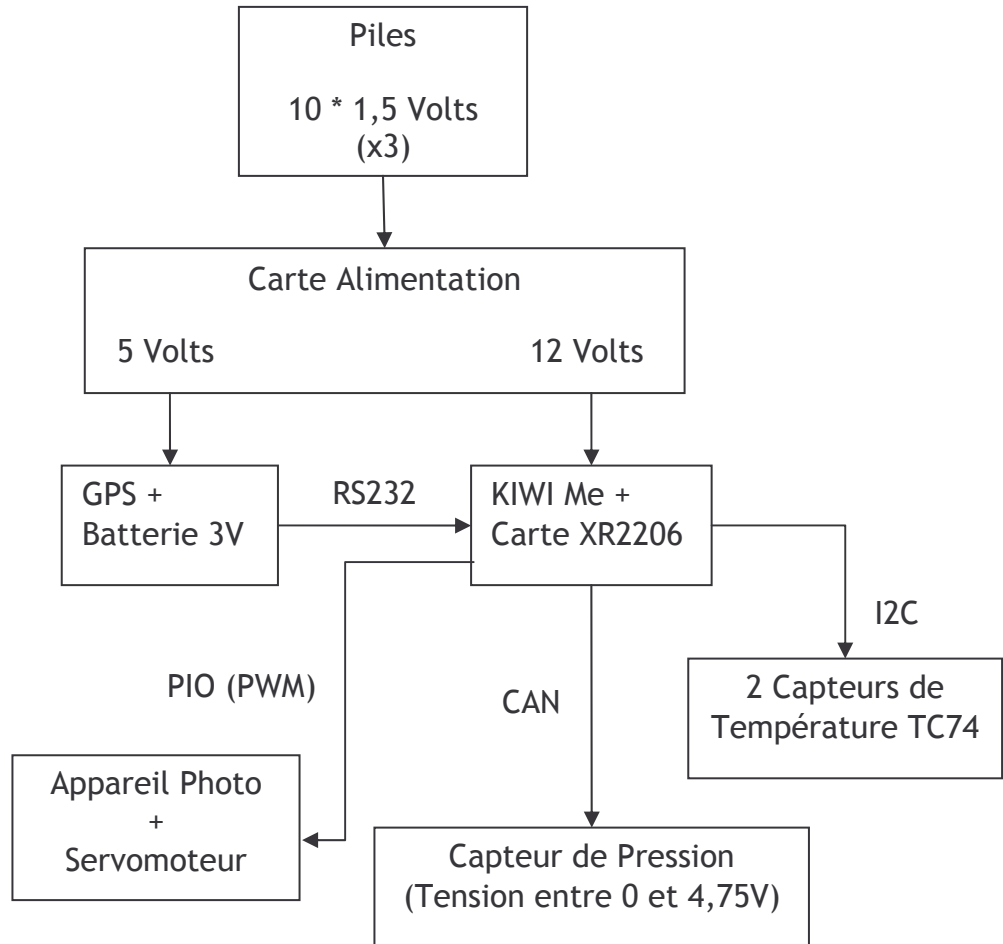
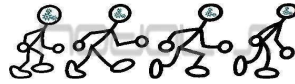


Figure 1. Schéma synoptique



2.2.1. Carte d'alimentation

La carte d'alimentation permet d'alimenter le GPS embarqué d'une part, et l'émetteur KIWI Me d'autre part. Cette carte est composée de deux alimentations à découpage Texas Instrument PT5100 (cf. **Figure 2**), auxquelles il a fallu rajouter 2 capacités sur chaque afin d'en assurer le correct fonctionnement. L'utilisation de ce type d'alimentation s'avère indispensable pour le bon fonctionnement du GPS, gros consommateur de courant.

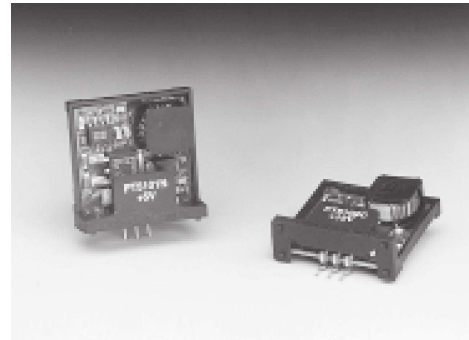


Figure 2. Photographie du PT5100

2.2.2. Emetteur KIWI Me et modulation FSK

L'innovation majeure du projet G2L réside dans la reprogrammation de l'émetteur KIWI Me encore jamais réalisée à ce jour (par un club espace). Le but de cette reprogrammation est d'utiliser l'émetteur comme seule CPU de tous les systèmes électroniques embarqués. Cette reprogrammation est réalisée grâce à l'ICD2 (programmeur de PIC de Microchip) et une connectique maison (cf. **Figure 3**).

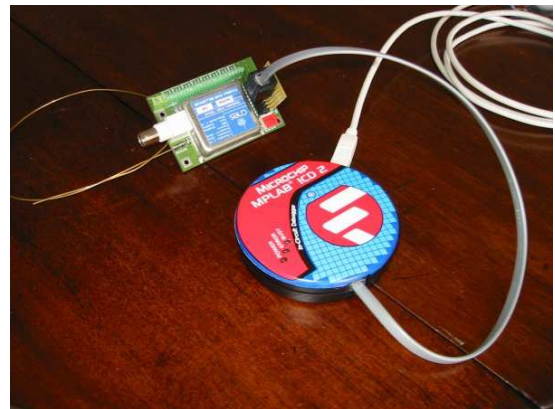


Figure 3. Reprogrammation du KIWI Me

Les différents capteurs utilisés sont reliés au PIC du KIWI Me qui génère alors des trames envoyées par une liaison série sur l'entrée FSK du composant XR2206 implanté sur un circuit imprimé externe conçu à cette seule fin. La fréquence FSK est générée par le circuit VCO (Voltage Controlled Oscillator) du XR2206. La sortie de ce dernier est connectée à la modulation externe du KIWI Me. Ainsi, l'émission des trames se fait par modulation FSK à un débit de 600 Bauds (cf. **Figure 4**).

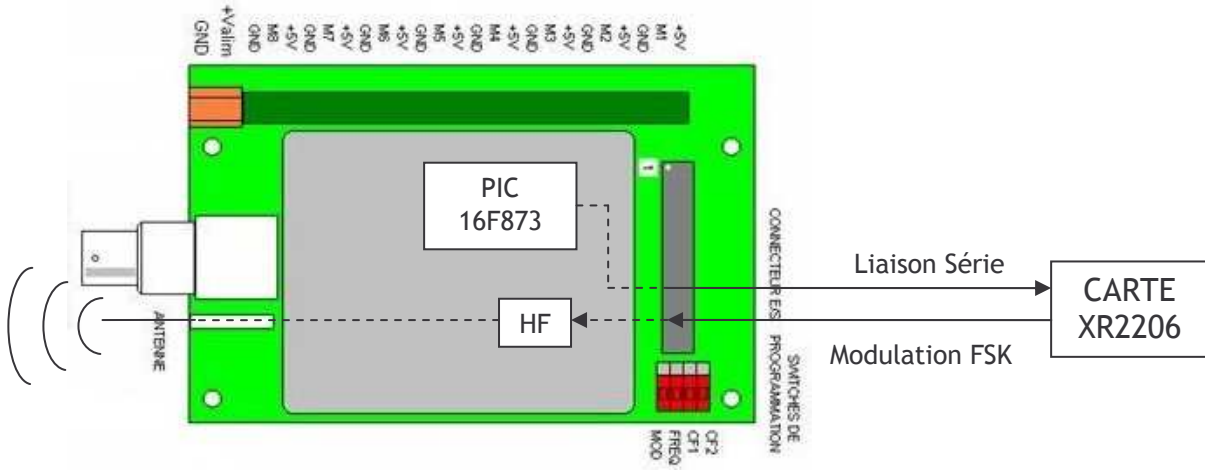


Figure 4. Représentation schématique de la liaison KIWI Me / Carte XR2206

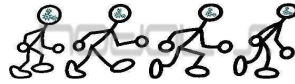
2.2.3. Capteurs de température interne et externe

La mesure des températures interne et externe est effectuée par des capteurs numériques TC74 (Microchip) alimentés par le KIWI Me en 5 Volts et communiquant avec l'émetteur par liaison I2C. En effet, nous avons remarqué que le bus I2C du PIC du KIWI Me était accessible via les commutateurs CF1 et CF2 du KIWI Me.

2.2.4. Capteur de pression

La mesure de la pression est réalisée grâce au capteur analogique MPX5100 (Motorola) relié au convertisseur analogique/numérique de l'émetteur KIWI Me. Bien que coûteux, il est très performant : il n'y a en effet besoin que de l'alimenter pour qu'il nous fournisse une information analogique sur la pression absolue variant entre 0 et 4,75V. Les données ainsi obtenues sont ensuite exploitées grâce à la table de conversion établie lors de l'étalonnage de ce capteur.

La tension analogique du capteur est reliée à la voie 1 du KIWI Me et nous faisons une conversion analogique/numérique directement par le PIC du KIWI Me (il faut faire attention de ne pas oublier de piloter le démultiplexeur des 8 voies habituelles du KIWI Me).



2.2.5. Système de positionnement

Le GPS présent dans le ballon est un GPS Motorola de la famille Oncore (cf. **Figure 5**). Nous communiquons avec lui par une interface série RS232.

Nous l'avons basculé en NMEA pour qu'il nous fournisse ses informations en code ASCII, afin de faciliter le debug.

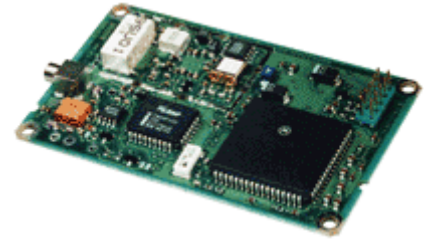


Figure 5. Photo du GPS Oncore

2.2.6. Système de prise de vues

La prise de vues est réalisée grâce à un appareil photo déclenché par un servomoteur. Ce dernier est piloté par le microcontrôleur du KIWI Me programmé pour récupérer l'altitude du GPS embarqué dans la nacelle. La prise de vue est déclenchée tous les 500m au cours de la phase ascendante du vol du ballon.

3. Suivi du ballon

3.1. Le logiciel de réception

Nous avons conçu un logiciel de réception afin de visualiser la télémétrie en temps réel. Il se compose de plusieurs vues (cf. **Figure 6**) :

- ✓ La fenêtre *ENVIRONNEMENT* permet de visualiser les températures interne et externe, une image de la pression en volts ainsi que le nombre de photos prises,
- ✓ La fenêtre *GPS VOITURE* nous informe sur la position de la voiture grâce à la longitude et la latitude fournies par un GPS à main connecté à un des ports série de l'ordinateur. Cette vue nous donne également la distance entre le ballon et la voiture (en kilomètres) ainsi que le cap à prendre par rapport au nord géographique,
- ✓ La fenêtre *GPS NACELLE* permet de visualiser la longitude, la latitude et l'altitude de la nacelle.



Figure 6. Capture d'écran du logiciel en fonctionnement

Les données reçues par le logiciel sont enregistrées dans des fichiers différents :

- ✓ Un fichier binaire qui enregistre les informations brut reçues,
- ✓ Un fichier csv (données mises en forme et séparées par des virgules) qui permet d'exporter et de traiter les informations plus facilement dans excel.

3.2. *La chasse au ballon*

Après le lâcher réalisé lors de la campagne nationale de lancement de Planète Sciences à La Courtine à 10h30 AM, par mauvais temps, nous avons suivi le ballon sur plus de 300 Kms, pendant près de 9h. Cela a été possible grâce au soft de réception de télémesure qui indique les positions du ballon et de la voiture ainsi que le cap à prendre et grâce au récepteur HF embarqué dans la voiture. La station au sol réalise les enregistrements des données des différents capteurs et des coordonnées du GPS embarqué.

Nous avons effectué un parcours quasiment parallèle à celui du ballon (cf. **Figure 7**), preuve que le soft de réception de télémesure est relativement fiable.

Malgré cela, le ballon n'a pu être retrouvé du fait de la dérive en température des composants. Ainsi, en fin de vol, nous n'arrivions plus à démoduler le signal envoyé par le ballon.

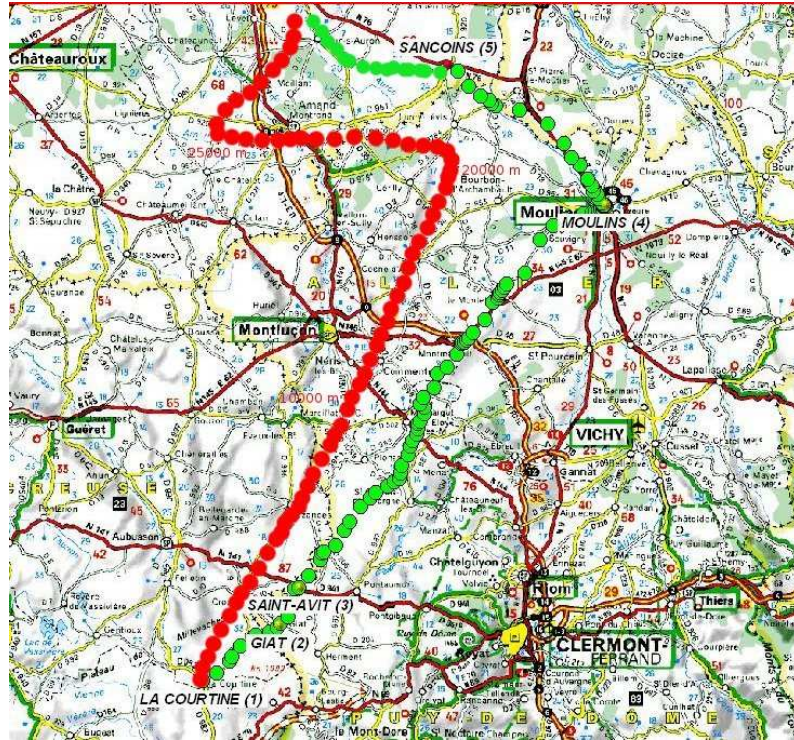


Figure 7. Parcours du ballon (rouge) et de la voiture (vert)

Au cours du gonflage de l'enveloppe, le copilote a préparé la chasse au ballon grâce aux informations recueillies le jour même, comme le sens du vent qui nous a donné la direction à prendre suite au lâcher. Nous avons cependant perdu du temps au démarrage, ce qui a entraîné un retard d'une demi heure sur le ballon. Afin de ne plus rencontrer ce genre de problèmes à l'avenir, il est conseillé de prendre contact avec Météo France afin d'obtenir la carte des vents, les températures pour préparer au mieux la chasse au ballon.

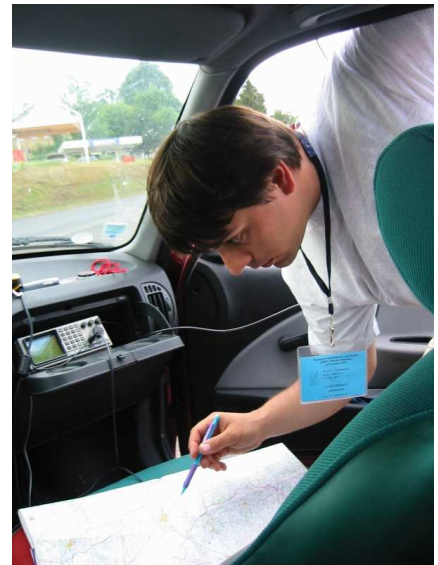


Figure 8. Manu lit dans les cartes



4. Résultats de la télémessure

Les résultats obtenus ne sont pas pleinement satisfaisants puisque le capteur de pression a saturé à partir de 15 000 m d'altitude et que la prise de vues n'a pas fonctionné. Cependant, les résultats présentés ci-dessous montrent que :

- ✓ La nacelle est bien isolée, avec une température interne qui ne descend jamais en dessous de 3 °C,
- ✓ La tropopause est à 15 000 m d'altitude le jour du lâcher,
- ✓ Les évolutions de la température interne et de la température externe sont assez semblables et correspondent aux données de la littérature.

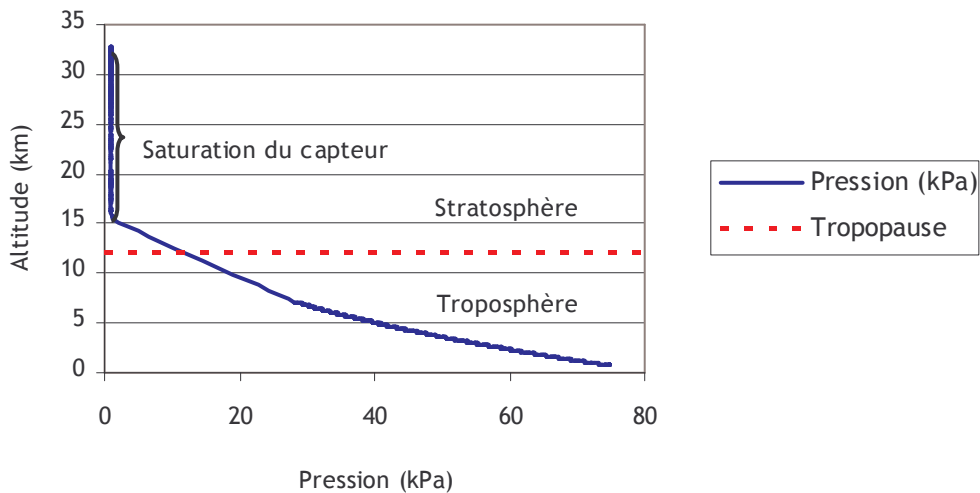


Figure 9. Pression mesurée par le ballon

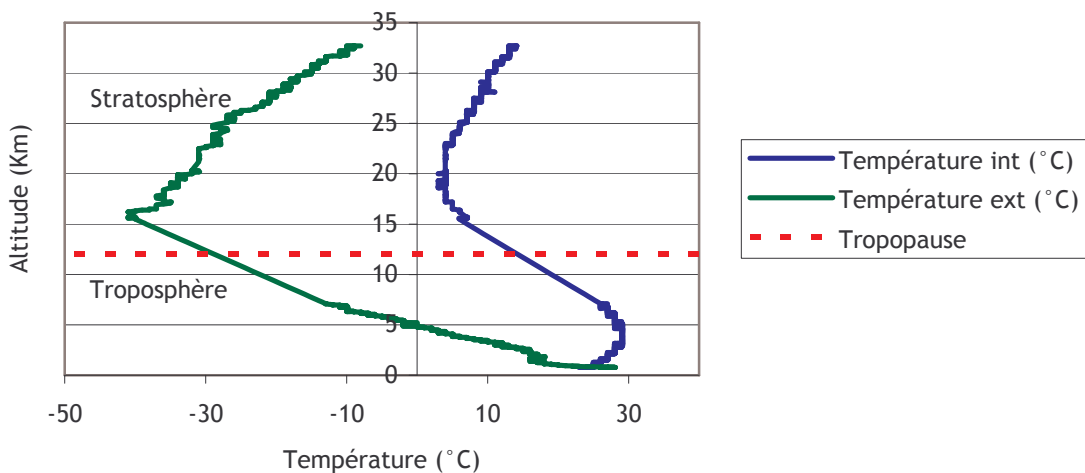


Figure 10. Températures mesurées par le ballon



5. Conclusion

C'est grâce à une forte volonté que ce projet a vu le jour et a pu être lâché après toute une année de travail, l'équipe n'ayant pu trouver de sponsors leur permettant de financer certaines des expériences embarquées.

Nous expliquons la non récupération du ballon par une préparation insuffisante de sa trajectoire, de la météo et de la chasse au ballon elle-même.

Le nouveau soft de réception de télémétrie écrit pour que Planète Sciences puisse l'exploiter à grande échelle par la suite n'est pas pleinement satisfaisant, c'est-à-dire que l'expérience devrait être reconduite sur un autre projet.

Malgré cela, les résultats obtenus sont très positifs puisqu'ils vont nous permettre de faire évoluer le soft de réception de télémétrie, la technique de chasse au ballon et d'aborder de nouvelles expériences à embarquer suite à cette aventure.

6. Remerciements

Un grand merci au **CNES**, à **Planète Sciences** et à tous ses **bénévoles** !

Un coup de chapeau tout particulier à **Etienne Maier** (responsable du secteur espace) pour son dynamisme, sa bienveillance et (ce serait trop long).

Merci aussi à :

DD pour son soutien technique sans faille ;)

Nico Couronneau pour le suivi,

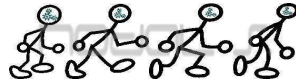
Manu Jolly pour sa lecture des cartes,

Laurent Vayssade pour le gonflage,

Florent Bouchoux pour la télémétrie,

Aline pour ses p'tites mains de bâcheuse hors paire,

Et **Natacha** pour la réalisation de la chaîne de vol (et ses câlins...)



7. Annexes

Une partie du code écrit pour G2L pouvant éventuellement intéresser d'autres clubs, en particulier la programmation de la PLL du KIWI ME Millenium, en voici une partie.

Embarque_G2L.c

```
1  /*
2   This file is part of NODALE'S / G2L.
3
4   G2L is free software; you can redistribute it and/or modify
5   it under the terms of the GNU General Public License as published by
6   the Free Software Foundation; either version 2 of the License, or
7   (at your option) any later version.
8
9   G2L is distributed in the hope that it will be useful,
10  but WITHOUT ANY WARRANTY; without even the implied warranty of
11  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
12  GNU General Public License for more details.
13
14  You should have received a copy of the GNU General Public License
15  along with G2L; if not, write to the Free Software
16  Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
17
18  Copyright 2004,2005 Olivier Sterenberg
19  */
20 /**
21  *   \file Embarque_G2L.c
22  *
23  *       Fichier principal du programme.
24  *
25  *   @version      0.2
26  *   @author       Olivier Sterenberg
27  *   @date         01/03/05
28  *
29  */
30 #include <pic.h>
31
32 #include "delayG2L.c"
33 #include "serieG2L.c"
34 #include "trameG2L.c"
35 #include "portG2L.c"
36 #include "gpsG2L.c"
37 #include "adcG2L.c"
38 #include "i2cG2L.c"
39 #include "frequenceG2L.c"
40
41 void main(void)
42 {
43     port_init();
44     serie_init();
45     adc_init();
46     i2c_init();
47     //gps_init();
48     SPBRG = ((int) (FOSC/(16UL*(4800-1))));
49
50     if (RC0==1)
51     {
52         PLL_1();           //inter vers le bas
53     }
54     else
55     {
56         PLL_2();           //inter vers le haut
57     }
```

Embarque_G2L.c

```
58
59     MyPWM(90);
60
61     while(1){
62
63         trameG2L[1] = temp_int();
64         trameG2L[2] = temp_ext();
65         trameG2L[3] = convert();
66
67         seuilTrameG2L();
68
69         prise_de_vue();
70
71         reponseGPS();
72         led_on();
73         sendTrameG2L(trameG2L, sizeof(trameG2L));
74         led_off();
75     }
76 }
77
78
```

```
1  /*
2   This file is part of NODALE'S / G2L.
3
4   G2L is free software; you can redistribute it and/or modify
5   it under the terms of the GNU General Public License as published by
6   the Free Software Foundation; either version 2 of the License, or
7   (at your option) any later version.
8
9   G2L is distributed in the hope that it will be useful,
10  but WITHOUT ANY WARRANTY; without even the implied warranty of
11  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
12  GNU General Public License for more details.
13
14  You should have received a copy of the GNU General Public License
15  along with G2L; if not, write to the Free Software
16  Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
17
18  Copyright 2004,2005 Olivier Sterenberg
19  */
20
21  #ifndef _FREQUENCE_H_INCLUDED
22  #define _FREQUENCE_H_INCLUDED
23
24  #define Data_PLL    RB0
25  #define LE_PLL      RB1
26  #define Clock_PLL   RB2
27
28  #define TRIS_DATA   TRISB0
29  #define TRIS_LE     TRISB1
30  #define TRIS_Clk    TRISB2
31
32  void envoi_PLL(char temp_PLL);
33  void envoi_PLL_3b(char temp_PLL);
34  void PLL_1(void);
35  void PLL_2(void);
36
37  #endif
38
```

frequenceG2L.c

```

1  /*
2  This file is part of NODALE'S / G2L.
3
4  G2L is free software; you can redistribute it and/or modify
5  it under the terms of the GNU General Public License as published by
6  the Free Software Foundation; either version 2 of the License, or
7  (at your option) any later version.
8
9  G2L is distributed in the hope that it will be useful,
10 but WITHOUT ANY WARRANTY; without even the implied warranty of
11 MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
12 GNU General Public License for more details.
13
14 You should have received a copy of the GNU General Public License
15 along with G2L; if not, write to the Free Software
16 Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
17
18 Copyright 2004,2005 Olivier Sterenberg
19 */
20 /**
21 * \file frequenceG2L.c
22 *
23 * Fonctions d'initialisation de la PLL du KIWI.
24 *
25 * @version 0.1
26 * @author Olivier Sterenberg
27 * @date 01/03/05
28 */
29
30 #ifndef _FREQUENCE_C_INCLUDED
31 #define _FREQUENCE_C_INCLUDED
32
33 #include "frequenceG2L.h"
34
35 /**
36 * Envoie du paramètre à la PLL
37 *
38 * MSB First !
39 *
40 * @param Carctère à envoyer sur la PLL
41 */
42 void envoi_PLL(char temp_PLL)
43 {
44     char DD;
45
46     for(DD=8; DD>0; DD--){
47         if(temp_PLL&0x80){
48             Data_PLL = 1;
49             Clock_PLL = 1;
50             Clock_PLL = 1;
51             Clock_PLL = 0;
52             Clock_PLL = 0;
53         }
54         else{
55             Data_PLL = 0;
56             Clock_PLL = 1;
57             Clock_PLL = 1;

```

frequenceG2L.c

```

58         Clock_PLL = 0;
59         Clock_PLL = 0;
60     }
61     temp_PLL <<= 1;
62 }
63 }
64
65 /**
66 * Envoie des 3 bits de poids fort du paramètre à la PLL
67 *
68 * @param 3 bits de poids fort du caractère passé à envoyer sur la PLL
69 */
70 void envoi_PLL_3b(char temp_PLL)
71 {
72     char DD;
73
74     for(DD=8; DD>5; DD--){
75         if(temp_PLL&0x80){
76             Data_PLL = 1;
77             Clock_PLL = 1;
78             Clock_PLL = 1;
79             Clock_PLL = 0;
80             Clock_PLL = 0;
81         }
82         else{
83             Data_PLL = 0;
84             Clock_PLL = 1;
85             Clock_PLL = 1;
86             Clock_PLL = 0;
87             Clock_PLL = 0;
88         }
89         temp_PLL <<= 1;
90     }
91 }
92
93 /**
94 * Fonction de configuration de la PLL sur la fréquence 1 du KIWI
95 *
96 *
97 */
98 void PLL_1(void) {
99     // FREQUENCE => 137,950 MHz
100
101     // P | R | C
102     // 1 | 640 | 1
103     // 1 | 0000101 0000000 | 1
104
105     #define P_R_11 0x85
106     #define P_R_21 0x01
107
108
109     // B | A | C
110     // 86 | 14 | 0
111     // 00001010 110 | 00011 10 | 0
112
113
114     #define B_A_11 0x0A

```

frequenceG2L.c

```

115 #define B_A_21 0xC3
116 #define B_A_31 0x80
117
118     LE_PLL = 1;
119     LE_PLL = 1;
120
121     Data_PLL = 1;
122     Clock_PLL = 1;
123     Clock_PLL = 1;
124     Clock_PLL = 0;
125     Clock_PLL = 0;
126     Data_PLL = 0;
127
128     LE_PLL = 0;
129     LE_PLL = 0;
130
131     envoi_PLL(P_R_11);
132     envoi_PLL(P_R_21);
133     LE_PLL = 1;
134     LE_PLL = 1;
135     LE_PLL = 0;
136     LE_PLL = 0;
137
138     envoi_PLL(B_A_11);
139     envoi_PLL(B_A_21);
140     envoi_PLL_3b(B_A_31);
141     LE_PLL = 1;
142     LE_PLL = 1;
143     LE_PLL = 0;
144     LE_PLL = 0;
145 }
146
147 /**
148  *   Fonction de configuration de la PLL sur la fréquence 2 du KIWI
149  *
150  *
151  */
152 void PLL_2(void) {
153     // FREQUENCE => 138,5 MHz
154
155     // P | R | C
156     // 1 | 640 | 1
157     // 1 | 0000101 0000000 | 1
158
159     #define P_R_12 0x85
160     #define P_R_22 0x01
161
162     // B | A | C
163     // 86 | 36 | 0
164     // 00001010 110 | 01001 00 | 0
165
166     #define B_A_12 0x0A
167     #define B_A_22 0xC9
168     #define B_A_32 0x00
169
170     LE_PLL = 1;
171     LE_PLL = 1;

```

frequenceG2L.c

```

172
173     Data_PLL = 1;
174     Clock_PLL = 1;
175     Clock_PLL = 1;
176     Clock_PLL = 0;
177     Clock_PLL = 0;
178     Data_PLL = 0;
179
180     LE_PLL = 0;
181     LE_PLL = 0;
182
183     envoi_PLL(P_R_12);
184     envoi_PLL(P_R_22);
185     LE_PLL = 1;
186     LE_PLL = 1;
187     LE_PLL = 0;
188     LE_PLL = 0;
189
190     envoi_PLL(B_A_12);
191     envoi_PLL(B_A_22);
192     envoi_PLL_3b(B_A_32);
193     LE_PLL = 1;
194     LE_PLL = 1;
195     LE_PLL = 0;
196     LE_PLL = 0;
197
198 }
199
200 #endif
201

```