

Séquenceur à micro contrôleur PIC16F

F. Douguet , CEEB, 2014

Pour quoi faire ?

L'intérêt d'un micro contrôleur dans un séquenceur ne va pas de soit. Pour s'engager dans une réalisation plus exigeante en terme de compétences et d'outils de conception ; il faut bien quelques avantages.

Signalisation sonore

La signalisation par diodes électroluminescentes (DEL) des phases de vol est très simple à mettre en oeuvre dans un séquenceur ; cependant voir les DEL s'éclairer sous le soleil du sud ouest est beaucoup plus difficile. Signaler les phases de vol par un buzzer émettant des séquences de bip courts et longs selon la phase identifiée est assez difficile à réaliser en composants discrets ou avec des portes logiques. Ce n'est pas impossible évidemment mais le nombre de composants peut rapidement devenir important et la flexibilité du système est très limité comme souvent avec l'utilisation de composants non programmables.

Détection de départ

La détection de décollage de la fusée par ouverture d'un contact normalement fermé est assez simple à réaliser. Si vous souhaitez utiliser un accéléromètre ; l'utilisation de composants non programmable est synonyme d'usine à gaz non modifiable. En effet le capteur va bien vous délivrer l'information mais il faut s'assurer que cette information soit durable ; qu'il ne s'agit pas d'un choc. Encore une fois c'est réalisable en composants discrets ; mais tellement plus simple avec un micro contrôleur !

Flexibilité et confort de mise au point

Un grand nombre de fonctions matériel des systèmes non programmables sont assurées par le soft de leur équivalent programmable. Là où il fallait dessouder un composant voir refaire un circuit imprimé ; il suffit de corriger le code source.

Comment le faire ?

Choix du micro contrôleur

Logiquement il est nécessaire de spécifier clairement vos besoins en terme d'entrées/sorties, de conversion analogique / numériques, de standard de communication (RS232, CAN, I2C...) et de vitesse d'exécution. Après cette étape vous partez à la chasse de la perle rare pas cher et facile à souder sur les sites des constructeurs : ATMEL, MICROCHIP, STMICRO, CYPRESS, et pourquoi pas XILINX, ALTERA, et autres fondeurs de puces programmables en tout genre !

Voulant laisser la plus grande marge de choix à une équipe souhaitant réaliser un séquenceur micro contrôlé ; je n'avais indiqué que le constructeur (MICROCHIP) . Quelques jours plus tard , ma correspondante m'a fait part de sa difficulté à choisir parmi les centaines de références existantes ! Je fus donc un peu plus prescripteur sur demande de l'équipe.

Un choix raisonnable est synonyme de facilité de mise en oeuvre. Les outils de programmation doivent être facile à se procurer et le moins cher possible. J'ai donc conseillé à l'équipe de se procurer un kit de développement comprenant la totalité des outils. Le PICKIT2 (Microchip) contient un programmeur avec une interface USB, une platine équipé d'un micro 16F690 ou 16F887 selon le kit commandé et un CD où vous trouverez le logiciel permettant de commander le programmeur et des logiciels de programmation assembleur et C. En bref la totale !

Vous pouvez vous procurez cet ensemble chez votre fournisseur préféré en tapant PICKIT2 dans la zone de recherche à un prix compris entre 40 et 50 € hors taxes.

Concernant le logiciel de programmation ; MIKROELEKTRONIKA propose gratuitement une version limité en taille de code mais avec toutes les bibliothèques de son compilateur MIKRO C PRO for PIC. C'est le choix pédagogique le plus raisonnable vu la quantité impressionnante de bibliothèques fournies.

Je pars donc sur une solution à base de PIC, le 16F690 et d'un code source en C écrit avec MIKRO C Pro for Pic de Mikroelektronika.

Installation des logiciels

L'installation ne pose aucune difficulté. Après lancement de PICKIT2 V2.6 ou plus récent, le logiciel doit détecter votre programmeur. Connectez votre platine sans l'alimenter puis cocher la case "ON" dans VDD PICkit 2 (à droite). Réglez la tension à 5V ; le programme inséré d'origine dans le PIC doit s'exécuter. **N'appliquez aucune alimentation extérieure.**

Le téléchargement et l'installation de la version gratuite du compilateur C de Mikroelektronika ne présente aucune difficulté. Une fois installé le compilateur vous permet de programmer et de compiler vos projets.

Programmation

Le programme est intimement lié à vos intentions et au schéma de votre platine finale. Pour aller rapidement vers une solution pratique je vous propose un schéma dont vous pourrez mettre en oeuvre, sur votre platine, les parties qui vous intéressent ; ainsi la progression sera la votre et à votre rythme.

Le Schéma

Séquenceur à micro contrôleur dans sa version 2

Ce séquenceur permet :

- de régler à l'aide de 5 micro interrupteurs la durée entre le départ et l'ouverture du parachute (SW1 à gauche de 1 à 31 sec),
- de commander l'ouverture des dispositifs de freinage, d'atterrissage si ce n'est pas le même et de libération du dispositif de freinage,
- d'accueillir l'information d'un capteur externe (CON CAP EXT et U4) et de paramétrer le fenêtrage (+ ou - 2 secondes autour de la durée nominale) grâce à CA1,
- d'émettre des signaux sonores (BUZ1),
- d'informer les autres systèmes de la fusée qu'elle est alimentée (U3, commande d'alimentation en cascade), qu'elle a décollée, que le frein est déployé puis libéré et finalement que l'atterrisseur est déployé (U2, U3, U5, U6, U7). Grâce aux optocoupleurs et conformément au cahier des charges FUSEX, ces informations sont diffusées en garantissant au séquenceur un isolement galvanique total avec les autres systèmes embarqués.

Bien évidemment chacun choisira de mettre en oeuvre les différentes parties du séquenceur : si vous n'avez qu'un parachute, supprimer les composants de commande de frein et libération de frein pour ne garder que ceux dédiés à l'atterrisseur. Si l'info "décollage" n'intéresse pas les autres systèmes de votre fusée; supprimer U2 et les composants associés.

Le programme

```
/* Séquenceur de vol CCEB */
#define T1    PORTB.B4
#define T2    PORTC.B2
#define T4    PORTC.B1
#define T8    PORTC.B0
#define T16   PORTA.B2
#define Buzzer PORTB.B5
#define Fenetre PORTB.B6
#define StartJack PORTC.B6
#define cde_frein PORTC.B4
#define cde_lib_frein PORTC.B3
#define cde_atterrissage PORTC.B5

#define tempo_frein 45 // durée pendant laquelle le (petit) parachute de freinage est déployé
#define tempo_accel 2  // durée entre la libération du parachute de freinage et celui d'atterrissage

unsigned int tempo=31,n=0;

void Init_Main() {
ANSEL = 0;    // Affectation des broches à la lecture de signaux analogique
ANSELH = 0;
C1ON_bit = 0; // Paramétrage des comparateurs des timer 1 et 2 (0 = off)
C2ON_bit = 0;

// Paramétrage des entrées sorties des ports A,B et C de notre micro
TRISA0_bit=1; //Entrée Capteur externe de déclenchement ouverture frein
TRISA2_bit=1; //Entrée numérique T16

TRISB4_bit=1; //Entrée numérique T1
TRISB5_bit=0; //Sortie BUZZER
TRISB6_bit=1; //Option déclenchement par capteur externe

TRISC0_bit=1; //Entrée numérique T8
TRISC1_bit=1; //Entrée numérique T4
TRISC2_bit=1; //Entrée numérique T2
TRISC3_bit=0; //Sortie de cmd de libération de la moitié des suspentes du parachute frein
TRISC4_bit=0; //Sortie de commande d'ouverture de trappe du parachute frein
TRISC5_bit=0; //Sortie de commande d'ouverture de trappe du parachute d'atterrissage
TRISC6_bit=1; //Entrée START JACK : détection du départ de la fusée
TRISC7_bit=0;

// quel niveau logique doit être présent sur la broche au moment du démarrage
cde_lib_frein=0;
cde_frein=0;
cde_atterrissage=0;
```

```

/* Oscillateur interne à 8MHz */
IRCF0_bit=1;
IRCF1_bit=1;
IRCF2_bit=1;
} //fin Init_Main()

/* Lecture des micro interrupteurs */
void lecture_dip(){
tempo=31;           //Au départ tempo est fixée à son maximum
if(T1) tempo = tempo-1; //Si T1 = 1 (T1 représentant une seconde sur off) , on soustrait 1 à tempo
if(T2) tempo = tempo-2; //Si T2 = 1 (T2 représentant deux secondes sur off) , on soustrait 2 à tempo
if(T4) tempo = tempo-4; //Si T4 = 1 (T4 représentant 4 secondes sur off) , on soustrait 4 à tempo
if(T8) tempo = tempo-8; //Si T8 = 1 (T8 représentant 8 secondes sur off) , on soustrait 8 à tempo
if(T16) tempo = tempo-16; //Si T16 = 1 (T16 représentant 16 secondes sur off) , on soustrait 16
}

void main(){
Init_Main();
Buzzer=0; //Buzzer sur off (pas de son) au cas où ...
lecture_dip(); //Lecture de la consigne de temporisation
GIE_bit=0; //bit d'autorisation des interruptions à 0 : aucune interruption

while(!StartJack){ //Tant que la jack est en place
Buzzer=1;
delay_ms(50); //Buzz pendant 50ms
Buzzer=0;
delay_ms(950); //Silence pendant 950ms
}

for (n=0;n<(tempo*2);n++){ //tempo*2 boucles de 500ms soit une durée de "tempo"
Buzzer=1;
delay_ms(50); //Buzz pendant 50ms
Buzzer=0;
delay_ms(450); //Silence pendant 450ms
}

if(!Fenetre){ // Si un capteur externe déclenche l'ouverture et donc que le contact "fenetre" est fermé
cde_frein=1; //Libération du frein
for (n=0;n<tempo_frein;n++){ // pendant tempo frein
Buzzer=1;
delay_ms(50); //Buzz 50ms
Buzzer=0;
}
}
}

```

```
delay_ms(375); // Silence 375ms
Buzzer=1;
delay_ms(200); //Buzz 200ms
Buzzer=0;
delay_ms(375); // Silence 375ms
} // Soit 1 seconde par boucle
cde_frein=0; // Arrêt de la commande du frein
cde_lib_frein=1; // Libération du parachute de freinage

for (n=0;n<(tempo_accel);n++){
Buzzer=1;
delay_ms(50);
Buzzer=0;
delay_ms(50);
Buzzer=1;
delay_ms(50);
Buzzer=0;
delay_ms(325);
Buzzer=1;
delay_ms(200);
Buzzer=0;
delay_ms(325);
} // Soit 1 seconde par boucle

cde_lib_frein=0;
} //Fin if(!fenetre)
cde_atterrissage=1; // Déploiement du parachute d'atterrissage

while(1){ // Tant que le séquenceur est alimenté
Buzzer=1;
delay_ms(500); // Buzz 0.5 sec
Buzzer=0;
delay_ms(4500); // Silence pendant 4.5 sec
}
}
```

Fichier hex permettant de programmer le micro (tempo freinage 45 sec - tempo accel 2 sec)

SDV.hex :

```
:020000007E2858
:10000600831603178C1C0928000003288312031388
:10001600260803178D000313270803178C00831681
:100026008C130C150B08F0008B1355308D00AA307D
:100036008D008C14F01B21288B1322288B170C1192
:020046000800B0
:0E0048008312031325088A0024088200080092
:100056001F3083120313A0000030A100061E3728AC
:100066000130A002031CA103071D3D280230A00297
:10007600031CA103871C43280430A002031CA10310
:10008600071C49280830A002031CA103051D4F28A0
:100096001030A002031CA103A6012008A70003201C
:0200A600080050
:1000A80024208A110A128000840AA40A0319A50AC6
:0800B800F003031D54280800A9
:1000C000831203179E019F0199139A13831603133A
:1000D000051405150616861206170714871407154A
:1000E00087110712871207178713831287110712C8
:0C00F000871283160F168F160F170800DA
:1000FC001F3083120313A000A101A201A3016020F1
:10010C00831286122B208B1383120313071BA9282F
:10011C0086168230FC00DD30FD00FD0B9328FC0BB5
:10012C0093280000000086120A30FB00A430FC006B
:10013C007F30FD00FD0BA028FC0BA028FB0BA0289A
:10014C00000000008A28A201A3012008F100210868
:10015C00F200F10DF20D711072082302031DB82884
:10016C00710822020318D82886168230FC00DD3074
:10017C00FD00FD0BBF28FC0BBF2800000000861201
:10018C000530FB009130FC00CF30FD00FD0BCC287E
:10019C00FC0BCC28FB0BCC2800000000A20A031996
:1001AC00A30AAB28061B78290716A201A30100306D
:1001BC002302031DE3282D30220203181F29861663
:1001CC008230FC00DD30FD00FD0BEA28FC0BEA2838
:1001DC000000000086120430FB00CF30FC00013020
:1001EC00FD00FD0BF728FC0BF728FB0BF728000094
:1001FC00000086160330FB000830FC007730FD0051
:10020C00FD0B0629FC0B0629FB0B06298612043074
:10021C00FB00CF30FC000130FD00FD0B1329FC0B63
:10022C001329FB0B132900000000A20A0319A30ACF
:10023C00DD2807128715A201A30100302302031D3C
:10024C002929023022020318772986168230FC00F5
:10025C00DD30FD00FD0B3029FC0B302900000000C7
:10026C0086128230FC00DD30FD00FD0B3B29FC0BBF
:10027C003B290000000086168230FC00DD30FD00BA
:10028C00FD0B4629FC0B46290000000086120430A9
:10029C00FB004D30FC002330FD00FD0B5329FC0B03
:1002AC005329FB0B532986160330FB000830FC0046
:1002BC007730FD00FD0B6029FC0B6029FB0B6029DE
:1002CC0086120430FB004D30FC002330FD00FD0B8A
:1002DC006D29FC0B6D29FB0B6D29A20A0319A30ACE
:1002EC0023298711871686160630FB001330FC0075
:1002FC00AD30FD00FD0B8029FC0B8029FB0B802908
:10030C000000000086122E30FB00A930FC002F30BC
:10031C00FD00FD0B8F29FC0B8F29FB0B8F297929F5
:02032C00962910
:02400E00F50FAC
:00000001FF
```