

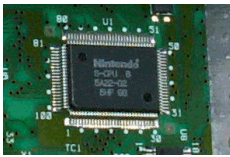
## Réalisez une interface clavier PC AT - Super-Oric ! (2/3)

par Thierry

### Protocole clavier Super-Oric

Fabrice a implémenté, dans la ROM BASIC du Super-Oric, un protocole de demande de transmission de l'état de la matrice de contacts du clavier Oric. Les chronogrammes associés, ainsi que les codes correspondants (extraits des ROM Super-Oric et PROM microcontrôleur 68HC705KJ1A), sont indiqués ci-après :

**Listing 1 : Routines extraites du code 65816 implémenté en ROM BASIC du Super-Oric, pour demander la transmission de l'état de la matrice de contacts du clavier Oric.**



Ce qui est fait à chaque retrace verticale de l'écran :

```
read_kbd
php
sep #$30 ; A,X,Y sur 8 bits

lda #1
sta $4016 ; reset pulse
jsr tempo ; 12 cycles
lda #0
sta $4016
jsr tempo2 ; 24 cycles = 6.3 µs

ldx #0
loopligne
ldy #8
loopcol
jsr tempo3 ; 48 cycles = 13.7 µs
lda $4017 ; + 15 cycles dans la boucle = 18 µs
lsr
ror $0320,x
dey
bne loopcol
lda $0320,x ; ajustement du décalage
asl
rol $0320,x
inx
cpx #8
bne loopligne

plp
rts

tempo3 ; 42 cycles
jsr tempo2
tempo2 ; 18 cycles
jsr tempo
tempo ; 6 cycles
rts
```

Ensuite, dans la rom Oric, le scan d'une colonne de clavier (routine \$F561) a été remplacé par:

```
and #7
sta $0211 ; ligne départ
testligne
tay
txa
eor #$FF
and $0320,y
bne foundkey
dey
tya
and #7
cmp $0211 ; a-t-on passé en revue toutes les lignes ?
bne testligne
lda #0
rts
foundkey
tya
ora #$80 ; renvoie la ligne et N=1
rts
```

**Listing 2 : Routine extraite du code 6805 implémenté en PROM du 68HC705KJ1A de l'interface, pour transmettre l'état de la matrice de contacts vers le Super-Oric.**

```
CLRXLDA #8
STA temp

; passer le top DATA LATCH du joystick

; attend le front descendant de DATA_LATCH
sm_bral BRSET DATA_LATCH,PORTA,*
BRCLR DATA_LATCH,PORTA,*
NOP ; nombre de NOP ajusté en cohérence
NOP ; avec la durée des tops de start
NOP ; joystick (12 µs) et clavier (8,4 µs).
NOP ; Les tests ont permis de déterminer
NOP ; une valeur de 5 à 7.

; attend le front descendant de DATA_LATCH
BRSET DATA_LATCH,PORTA,sm_bral

; attention : ROR détruit la matrice
sm_loop ROR ORIC_MATRIX_KBD,X
BCS sm_bra2

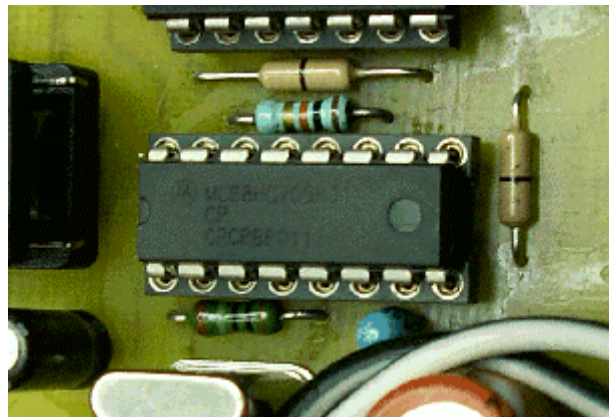
; 0 dans la matrice --> 1 sur SERIAL_DATA
BSET SERIAL_DATA,PORTA
BRA sm_bra3

sm_bra2 BCLR SERIAL_DATA,PORTA ; et inversement

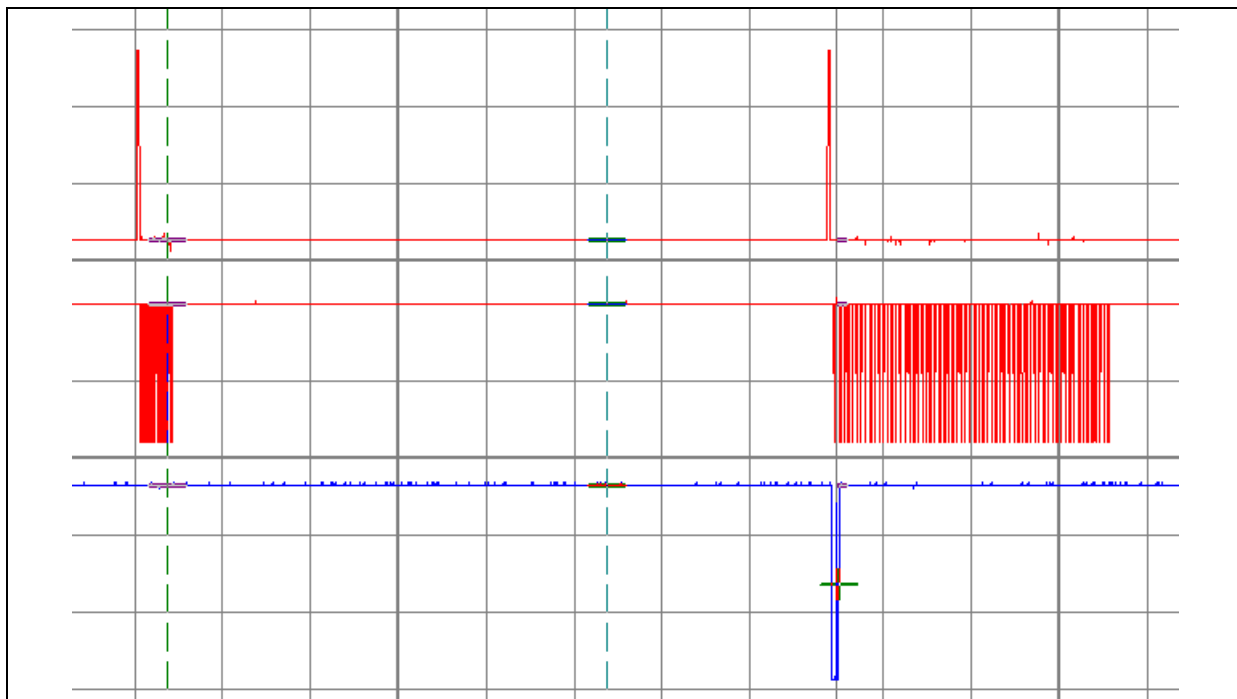
; attend le front descendant de DATA_CLOCK
sm_bra3 BRSET DATA_CLOCK,PORTA,*

DEC temp
BNE sm_loop
LDA #8 ; réinitialisation colonne
STA temp
INX
CPX #8
BNE sm_loop

; mise à zéro de la ligne pour scrutation par SNES
; en dehors de la routine d'envoi
BSET SERIAL_DATA,PORTA
```



Le code source, pour le microcontrôleur, a été diffusé sur l'eDisk de juin 2006 (version v1z3 pour ROM Super-Oric  $\geq 1.35$ ). Comme pour l'interface clavier PC AT – unité centrale Oric, la programmation du 68HC705KJ1A nécessite une circuiterie électronique spécifique : Je vous propose de programmer pour vous les microcontrôleurs dont vous auriez besoin, et de vous les expédier (ou de les apporter aux visus selon le calendrier)...



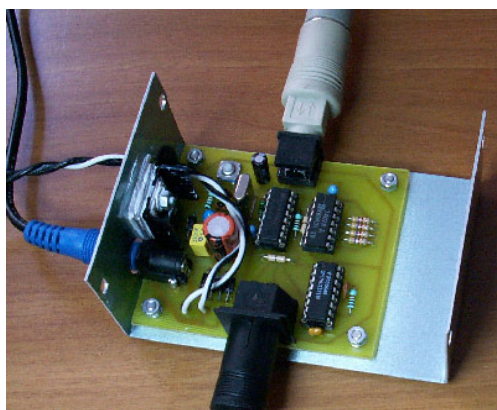
**Chronogramme 1 (2V/div., 500 microsecondes/div.) :**

Ce chronogramme montre que le microcontrôleur fait la distinction entre le créneau de start du joystick sur DATA\_LATCH (à gauche), et celui destiné au clavier (à droite). Cette distinction est basée uniquement sur la longueur de l'impulsion DATA\_LATCH :

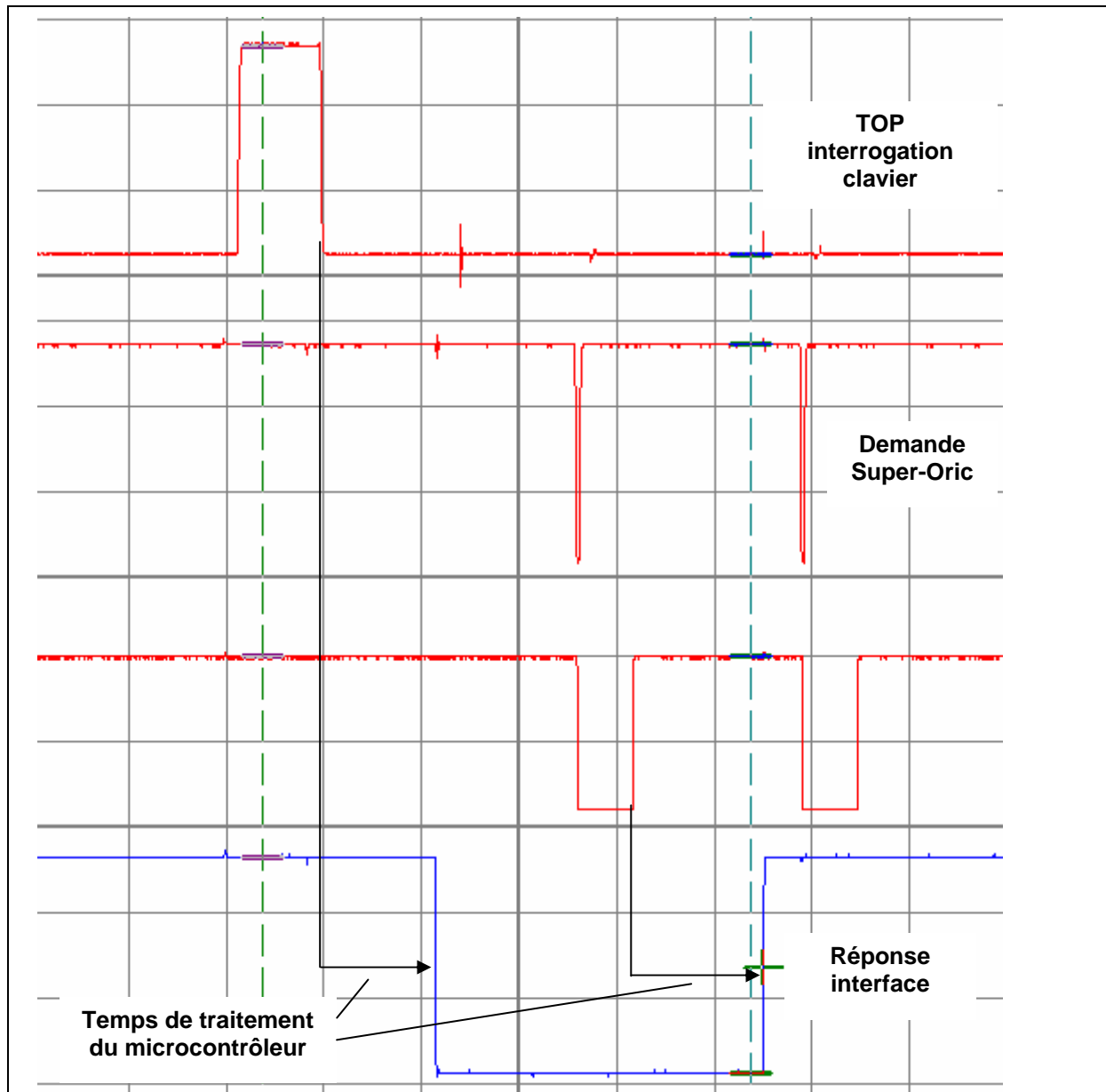
En haut, le signal DATA\_LATCH.

Au milieu, DATA\_CLOCK, remis en forme par le 74LS221 pour le microcontrôleur. La série d'impulsions, à gauche, est destinée au joystick (16 impulsions). Celle de droite est destinée au clavier (64 impulsions).

En bas, SERIAL\_DATA. L'impulsion sur SERIAL\_DATA correspond ici à la touche N (colonne 1 ligne 0 sur la matrice clavier Oric).



**Photo 1 : Détails de la mise en coffret.**



Chronogramme 2 (2V/div., 10 microsecondes/div.) :

Touche N enfoncée : De haut en bas :

- **DATA\_LATCH** (impulsion pour le clavier).
- **DATA\_CLOCK**.
- les impulsions **DATA\_CLOCK** remises en forme par le 74LS221, pour reconnaissance par le microcontrôleur. En effet, le signal **DATA\_CLOCK** d'origine est trop court pour être correctement identifié par le microcontrôleur.
- l'impulsion envoyée sur **SERIAL\_DATA** par le microcontrôleur, déclenché à partir de l'impulsion **DATA\_CLOCK** ou **DATA\_LATCH** clavier précédentes.

à suivre...