feux tricolores
en assembleur PIC

Andrew Pratt (Royaume-Uni)

Voici la programmation en assembleur PIC d'un ensemble de feux de circulation composé de six LEDs pour représenter les deux ensembles de feux rouge-jaune-vert (**fig. 1**). Le tout imite la signalisation des chantiers routiers pour le trafic alterné sur une seule voie.
À la mise sous tension, les deux feux seront au rouge pendant 10 s. Ensuite, le feu A passe au rouge et au jaune pendant 2 s puis au vert. Au bout de 20 s, le feu A passe au jaune pendant 5 s, puis au rouge. Les deux jeux de lumières resteront au rouge pendant 10 s avant que le feu B ne reproduise la séquence. Ce cycle se répétera jusqu'à ce que l'alimentation soit coupée. En plus de cela, nous avons un signal de validation générale de la séquence lumineuse - si le niveau de ce signal est bas, la séquence s'arrête au double rouge suivant. Le signal d'activation doit être haut pendant toute la période de 10 s au cours de laquelle les deux ensembles sont au rouge avant le début de la séquence alternée.
Cette description nous permet d'identifier les huit états possibles du système :

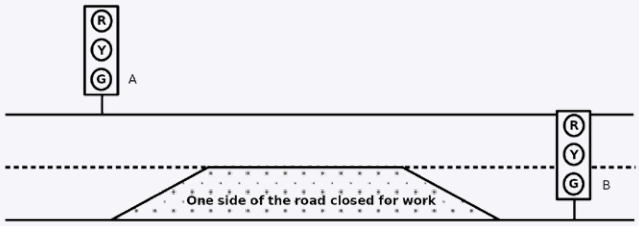

Figure 1. Commande de circulation alternée sur chantier routier. Remarquez le britannique sens de circulation.

état	feux A	feux B
0	rouge	rouge
1	rouge - jaune	rouge
2	vert	rouge
3	jaune	rouge
4	rouge	rouge
5	rouge	rouge-jaune
6	rouge	vert
7	rouge	jaune

La prochaine étape consiste à définir les transitions autorisées et les conditions d'entrée pour déclencher ces conditions. Il s'agit d'un exemple très simple dans la mesure où les états se succèdent les uns après les autres avec une seule transition autorisée vers l'état suivant. Le diagramme qui en résulte au format de la machine de Moore est présenté à la **figure 2**. Les sorties ne dépendent que de l'état en cours et sont indiquées sur le diagramme à l'intérieur de la boîte pour chaque état.
La transition vers l'état suivant est déterminée par le temps écoulé dans un état particulier, à l'exception des états 0 et 4 où il y a deux rouges. Dans ces états, le signal de validation doit être présent pendant 10 s. Cela couvre notre exigence de poursuite de la séquence vers les deux rouges suivants si le signal de validation est bas. Dans le schéma de la **figure 3,** six LED sont connectées par des résistances de limitation de

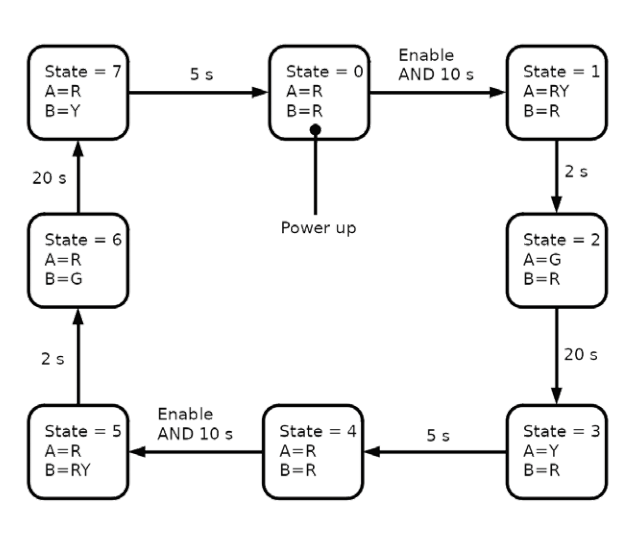

Figure 2. Carte de l'état des travaux routiers.

Listage 1. Programme complet pour les feux de circulation

```
;PROG_8_03.asm
LIST     P=16F1823
#INCLUDE <p16f1823.inc>
#INCLUDE <fsm_macros.inc>

    RADIX DEC              ; nombres en base 10 par défaut
    BOOK_CONFIGURATION     ; voir macro dans fsm_macros.inc

    CBLOCK 0x70
    TICKS                  ; le compte d'unités de 8,2ms
    SECONDS                ; le compte de secondes
    ENDC

    ORG 0X00
    GOTO START

    ORG 0X04               ; vecteur d'interruption
    BCF INTCON, TMR0IF     ; r.à.z. indicateur de dépassement tmr0
    DECF TICKS, F          ; décrémente TICKS de 1
    BTFSS  STATUS, Z       ; vérifie si TICKS=0
    GOTO $+4               ; si TICKS ≠0 retour d'interruption
    INCF SECONDS, F        ; si TICKS=0, incrémente les secondes
    MOVLW 122
    MOVWF TICKS            ; remet TICKS à 122 (122 x 8.2 ms 1 second).
    RETFIE                 ; Retour d'interruption

START
    MOVLB 1                ; Bank 1 requis pour les macros suivantes, et TRISC
    SET_FREQ_32MHZ         ; voir fichier fsm_macros.inc
    SET_TMR0_CASE1         ; donne 8,2 ms ticks. Voir fsm_macros.inc
    CLRF TRISC             ; PORTC en sortie
    MOVLB 0                ; bank 0 pour PORTA et PORTC
;*****************************************************************************************
S0
    MOVLW b'00100100'
    MOVWF PORTC                    ; allume rouge A (RC5) et rouge B (RC2), le reste éteint
SS0_0
    CLRF SECONDS                   ; efface le compteur de secondes
SS0_1
    BTFSS PORTA, 5
    GOTO SS0_0                     ; si le signal de validation est bas, retour à substate 0
    MOVLW 10
    IF_REG_LESS_THAN_W SECONDS
    GOTO SS0_1                     ; si moins de 10 s ont passé, rester dans substate 1
;*****************************************************************************************
S1
    MOVLW b'00110100'
    MOVWF PORTC             ; allume A rouge (RC5) A jaune (RC4) et B rouge (RC2), le reste éteint
SS1_0
    CLRF SECONDS            ; efface le compteur de secondes
SS1_1
    MOVLW 2
    IF_REG_LESS_THAN_W SECONDS
    GOTO SS1_1
S2
    MOVLW b'00001100'
    MOVWF PORTC                ; allume A vert (RC3) et B rouge (RC2), le reste éteint
SS2_0
    CLRF SECONDS
SS2_1
    MOVLW 20
    IF_REG_LESS_THAN_W SECONDS
    GOTO SS2_1                 ; si < 20 s, reste dans substate 1
```

suite au verso ...

```
;*****************************************************************************************
S3
    MOVLW b'00010100'
    MOVWF PORTC                     ; allume A jaune (RC4) et B rouge (RC2), le reste éteint
SS3_0
    CLRF SECONDS                    ; efface le compteur de secondes
SS3_1
    MOVLW 5
    IF_REG_LESS_THAN_W SECONDS
    GOTO SS3_1                      ; si < 5 s, reste dans substate 1
;*****************************************************************************************
S4
    MOVLW b'00100100'
    MOVWF PORTC                     ; allume A rouge (RC5) et B rouge (RC2), le reste éteint
SS4_0
    CLRF SECONDS                    ; efface le compteur de secondes
SS4_1
    BTFSS PORTA, 5
    GOTO SS4_0                      ; si le signal de validation est bas, retour à substate 0
    MOVLW 10
    IF_REG_LESS_THAN_W SECONDS
    GOTO SS4_1                      ; si < 10 s, reste dans substate 1
;*****************************************************************************************
S5
    MOVLW b'00100110'
    MOVWF PORTC             ; allume A rouge (RC5) et B rouge (RC2) et B jaune (RC1), le reste éteint
SS5_0
    CLRF SECONDS                ; efface le compteur de secondes
SS5_1
    MOVLW 2
    IF_REG_LESS_THAN_W SECONDS
    GOTO SS5_1                  ; si < 2 s, reste dans substate 1
;*****************************************************************************************
S6
    MOVLW b'00100001'
    MOVWF PORTC                   ; allume A rouge (RC5) et B vert (RC0), le reste éteint
SS6_0
    CLRF SECONDS                  ; efface le compteur de secondes
SS6_1
    MOVLW 20
    IF_REG_LESS_THAN_W SECONDS
    GOTO SS6_1                    ; si < 20 s, reste dans substate 1
;*****************************************************************************************
S7
    MOVLW b'00100010'
    MOVWF PORTC                 ; allume A rouge (RC5) et B jaune (RC1), le reste éteint
SS7_0
    CLRF SECONDS                ; efface le compteur de secondes
SS7_1
    MOVLW 5
    IF_REG_LESS_THAN_W SECONDS
    GOTO SS7_1                  ; si < 5 s, reste dans substate 1
    GOTO S0

    END
```

courant. La valeur de ces résistances n'est pas critique, 1 kΩ donnera environ 3 mA. Un PIC de type 16F823 est utilisé et il est programmé avec une passerelle FTDI USB/série.

Avant de produire le code qui fait fonctionner l'automate fini (= *state machine* en anglais) ci-dessus, le PIC doit avoir effectué certaines configurations, comme les paramètres du fichier de configuration, la fréquence de l'oscillateur, puis configuré les ports. Il est préférable de progresser par petits pas et de tester au fur et à mesure. Ces tests consistent à ajouter du code de débogage pour vérifier que chaque morceau de code ajouté fonctionne de façon fiable. N'essayez pas d'écrire un programme complet pour tenter ensuite de le faire fonctionner, vous vous y perdriez.

La **figure 4** est l'automate fini de haut niveau du programme en assembleur, auquel correspond le code assembleur PIC réel (**listage 1**). Le

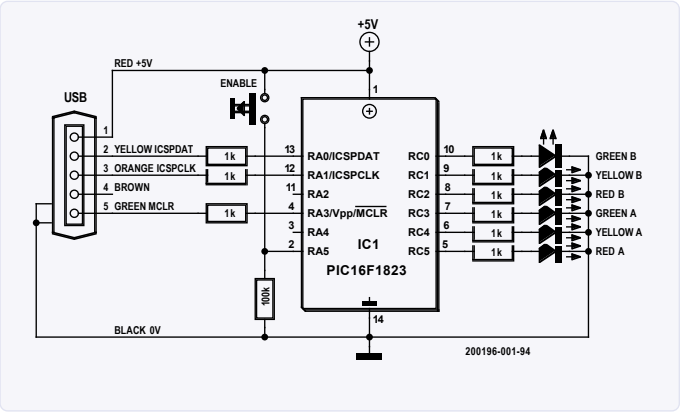

Figure 3. Schéma du système (virtuel) de commande des feux.

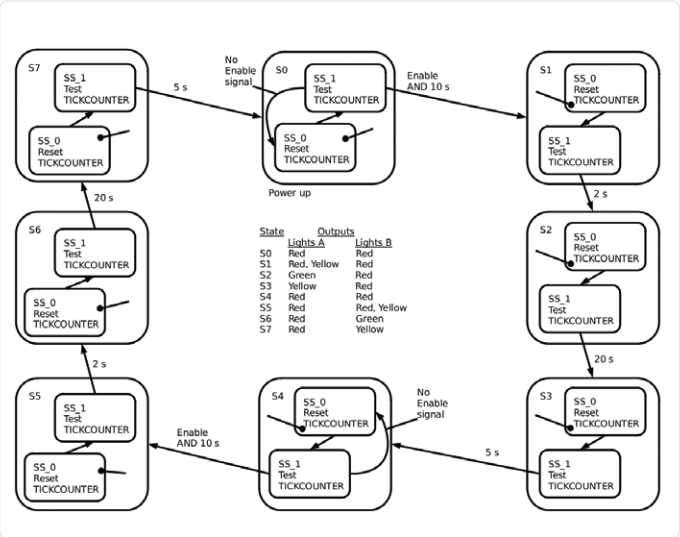

Figure 4. Tableau d'état complet pour les feux de circulation.

listage complet et la description du chronométrage et la vérification des entrées/sorties dépassent malheureusement le cadre de cet article, et il en va de même pour un programme spécial de test au démarrage et un mini-débogueur pour le compteur interne de secondes. Les deux derniers sont utiles pour construire le programme final de manière progressive et pédagogique. Tout cela est téléchargeable intégralement (diagrammes d'état compris) et gratuitement [1].

200196-04

LIEN

[1] **page de cet article** :
www.elektormagazine.fr/200196-03