```
// Dit programma kan geprogrammeerd worden in een PIC 12F683
// en heeft als doel: het op afstand in- en uitschakelen van
// navigatieverlichting, landingslichten en lichtbakens
// op een RC vliegtuig.
//
// Ruben Buysschaert, 19 januari 2011.
//
// www.rubu.be

#include "int16CXX.h"           //Deze header is nodig om enkele standaard routines
                                //voor tijdens de interrupt te kunnen gebruiken.
                                //Bijvoorbeeld: int_save_registers
#include "math16.h"

//Configuration word instellingen
#pragma config |= 0x00D4        // 0000.0000.1101.0100 = 0x00D4 gebruik
                                // interne oscillator van 8Mhz

//Definities van de in- en uitgangen
#pragma bit ch1                 @ GPIO.5        //input kanaal 1
#pragma bit ch2                 @ GPIO.4        //input kanaal 2
#pragma bit navigationLights    @ GPIO.0        //output
#pragma bit landingLights       @ GPIO.1        //output
#pragma bit beaconLights        @ GPIO.2        //output


//Prototypes van de functies
void init(void);
void initTimer1(void);          //Let op: deze functie werkt met Timer 0!
void Timer1On(void);
void Timer1Off(void);
void initTimer2(void);          //Let op: deze functie werkt met Timer 1!
void Timer2On(void);
void Timer2Off(void);
```

```
//Declaraties van de variabelen
char temp;
uns16 teller;

//Interrupt definitie
#pragma origin 4
interrupt interruptSubroutine(void)
{
    int_save_registers          //zie int16CXX.h

    //Interrupt van GPIO's (zowel bij dalende als bij stijgende flank)
    if(GPIF == 1)
    {
        temp = GPIO;            //GP lezen zodat de oude info overschreven
                                //wordt met de huidige info
        GPIF = 0;               //IF reseten

        //Indien kanaal 1 een signaal bevat, start de Timer
        if((ch1 == 1) && (T0IE == 0))
            Timer1On();         //start de meting van 1,5ms

        //Indien kanaal 2 een signaal bevat, start de Timer
        if((ch2 == 1) && (TMR1IE == 0))
            Timer2On();         //start de meting van 1,5ms
    }

    //Interrupt van timer 0, voor kanaal 1
    if((T0IF == 1) && (T0IE == 1))
    {
        T0IF = 0;
        Timer1Off();

        //na 1,5ms check ch1
        if(ch1 == 1)
            navigationLights = 1;
```

```
        else
            navigationLights = 0;
    }


    //Interrupt van timer 1, voor kanaal 2
    if((TMR1IF == 1) && (TMR1IE == 1))
    {
        TMR1IF = 0;
        Timer2Off();

        //na 1,5ms check ch2
        if(ch2 == 1)
            landingLights = 1;
        else
            landingLights = 0;
    }

    int_restore_registers
}


//Start van de main functie
void main( void)
{
    init();

    // Alle LED's uitschakelen
    navigationLights = 0;
    nop();
    landingLights = 0;
    nop();
    beaconLights = 0;

    while(1)
```

```
    {
        //Maak een 'flikker-sequentie' voor de lichtbakens,
        //indien de navigatielichten ingeschakeld zijn.
        if(navigationLights == 1)
        {
            teller++;
            if(teller==0)
                beaconLights = 1;
            if(teller==5000)
                beaconLights = 0;
            if(teller==7500)
                beaconLights = 1;
            if(teller==12500)
                beaconLights = 0;
        }
        else
            beaconLights = 0;

    }
}

void init(void)
{
    OSCCON.6 = 1;           //kies 8MHz
    OSCCON.5 = 1;
    OSCCON.4 = 1;

    CM2 = 1;                //comparatoren uitschakelen
    CM1 = 1;
    CM0 = 1;

    ANSEL = 0;              //Analoge ingangen uitschakelen

    TRISIO = 0b00111000;    //input-output instellen
```

```
    GIE = 1;                //interrupts toelaten
    GPIE = 1;               //interrupts van GPIO's toelaten
    IOC = 0b00110000;       //interrupts van pin 5 en 4 toelaten (ch1 en ch2)
    PEIE = 1;               //interrupts van pheriperals toelaten

    initTimer1();
    initTimer2();
}

void initTimer1()
{
    T0CS = 0;               //interne klok selectern
    PSA = 0;                //prescaler gebruiken
    PS0 = 0;                //instellen op 1:32
    PS1 = 0;
    PS2 = 1;
    TMR0 = 0;               //teller resetten
    T0IF = 0;               //IF clearen
    T0IE = 0;               //interrupt nog niet toelaten
}

void Timer1On()
{
    // Je wil 1,5ms meten. Aan een kloksignaal van 8MHz/4 = 2MHz => 0,5µs per tel,
    // met prescaler van 32 => één tel van Timer 0 = 0,5µs * 32 = 16µs
    // 1,5ms/16µs = 93,75 pulsen ~ 94 pulsen
    // 256 - 94 = 162 als init waarde.
    TMR0 = 162;             //teller resetten
    T0IF = 0;               //IF clearen
    T0IE = 1;               //interrupt toelaten
}

void Timer1Off()
{
    T0IE = 0;               //interrupt niet meer toelaten
```

```
    T0IF = 0;               //IF clearen
    TMR0 = 0;               //teller resetten
    //Let op: De timer is niet uitgeschakeld! Vandaar dat in de interrupt
    //subroutine, ook gecontrolleerd moet worden of T0IE == 1 ...
}


void initTimer2()
{
    TMR1CS = 0;             //interne klok selecteren
    T1CKPS1 = 1;            //prescaler op 1:8
    T1CKPS0 = 1;
    T1GE = 0;
    TMR1H = 0;              //teller resetten
    TMR1L = 0;
    TMR1ON = 1;             //timer inschakelen
    TMR1IF = 0;             //interrupt flag resetten
    TMR1IE = 0;             //interrupt nog niet toelaten
}

void Timer2On()
{
    // Je wil 1,5ms meten. Aan een kloksignaal van 8MHz/4 = 2MHz => 0,5µs per tel
    // met prescaler van 8 => één tel van Timer 0 = 0,5µs * 8 = 4µs
    // 1,5ms/4µs = 375 pulsen
    // 65536 - 375 = 65161 als init waarde
    // = 11111110.10001001
    TMR1L = 0b10001001;
    TMR1H = 0b11111110;
    TMR1IF = 0;
    TMR1IE = 1;             //interrupt toelaten
}

void Timer2Off()
{
```

```
    TMR1IE = 0;             //interrupt niet meer toelaten
    TMR1IF = 0;
    TMR1H = 0;
    TMR1L = 0;
    //Let op: De timer is niet uitgeschakeld! Vandaar dat in de interrupt
    //subroutine, ook gecontrolleerd moet worden of TMR1IE == 1 ...
}
```