```
//Dit programma kan geprogrammeerd worden in een PIC 16F88
//en heeft als doel de optimale verlichting van een Babboe bakfiets.
//Ruben Buysschaert, 1 november 2009.

#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"
//#include "16F88.h"    //optioneel

//Configuration word instellingen
#pragma config |= 0x3F18        // 11.1111.0001.1000 = 0x3F18 gebruik
                                // interne oscillator van 8Mhz

//Prototypes van de functies
void init(void);
void initTimer1(void);
void Timer1On(void);
void Timer1Off(void);

//Declaraties van de variabelen
char index;
uns16 sequentie1, sequentie2, sequentie3, sequentie4;
uns16 sequentie5, sequentie6, sequentie7, sequentie8;
bit klokpuls;

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

    if(TMR1IF == 1)         //interrupt van Timer1
    {
        TMR1IF = 0;
        if(index < 15)
            index++;
        else
            index = 0;
```

```
        TMR1H = 0b11000000; //laad de timer met een waarde

        klokpuls = 1;
    }

    if(INT0IF == 1)
        INT0IF = 0;         //Optie, wiel interrupt en dus wake up from sleep...

    int_restore_registers
}

//Definities van de in- en uitgangen
#pragma bit out1        @ PORTA.7
#pragma bit out2        @ PORTA.6
#pragma bit out5        @ PORTB.1
#pragma bit out6        @ PORTA.4
#pragma bit out7        @ PORTA.3
#pragma bit out8        @ PORTA.2
#pragma bit LDR         @ PORTA.1
#pragma bit schemer     @ PORTA.0
#pragma bit extra1      @ PORTB.5
#pragma bit extra2      @ PORTB.4
#pragma bit out4        @ PORTB.3
#pragma bit out3        @ PORTB.2
#pragma bit knop        @ PORTA.5
#pragma bit wiel        @ PORTB.0       //interrupt 0


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

    klokpuls = 0;

    while(1)
    {
        if(klokpuls == 1)
```

```
        {
            if(index == 0)
            {
                //Bepaal de sequenties van de uitgangen. Eén sequentie bestaat uit 16 stappen
                sequentie1 = 0b0101011111111111;        //koplamp links
                sequentie2 = 0b1111111111101010;        //koplamp rechts
                sequentie3 = 0b0010100000101000;        //zijpaneel links en rechts
                sequentie4 = 0b1111111111111111;        //reclame links en rechts
                sequentie5 = 0b0000000000101011;        //achteraan links
                sequentie6 = 0b0000010101000001;        //achteraan midden
                sequentie7 = 0b1010100000000001;        //achteraan rechts
                sequentie8 = 0b0000000000000000;        //uitgang voor het sturen van een optie...
            }
            out1 = sequentie1.0;
            sequentie1 = sequentie1 >> 1;
            out2 = sequentie2.0;
            sequentie2 = sequentie2 >> 1;
            out3 = sequentie3.0;
            sequentie3 = sequentie3 >> 1;
            out4 = sequentie4.0;
            sequentie4 = sequentie4 >> 1;
            out5 = sequentie5.0;
            sequentie5 = sequentie5 >> 1;
            out6 = sequentie6.0;
            sequentie6 = sequentie6 >> 1;
            out7 = sequentie7.0;
            sequentie7 = sequentie7 >> 1;
            out8 = sequentie8.0;
            sequentie8 = sequentie8 >> 1;

            klokpuls = 0;       //wacht op de volgende puls
        }
    }
}

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

```
    OSCCON.4 = 1;

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

    ANSEL = 0;

    GIE = 1;    //interrupts toelaten
    INT0IE = 1; //wiel interrupt toelaten

    TRISB  = 0b11110001;    //IO's instellen
    TRISA  = 0b00100011;    //RA5 is enkel input...

    initTimer1();
    Timer1On();
}

void initTimer1()
{
    TMR1CS = 0;         //internal clock as driver
    T1CKPS1 = 1;        //prescaler op 1:8
    T1CKPS0 = 1;
    T1OSCEN = 0;        //gebruik als timer, niet als counter
    TMR1ON = 0;         //timer nog niet inschakelen
    TMR1H = 0;          //teller resetten
    TMR1L = 0;
    TMR1H = 0b11000000; //laad de timer met een waarde
    TMR1IF = 0;         //interrupt flag resetten
    TMR1IE = 1;         //interrupt toelaten
    PEIE = 1;           //interrupts toelaten
    GIE = 1;            //idem
}

void Timer1On()
{
    TMR1ON = 1;
}
```

```
void Timer1Off()
{
    TMR1ON = 0;
}
```