

```

#include "config.h"
#include <xc.h>
#include <pic16f688.h>

//Zimmerbeleuchtung für das Haus
#define RAUM_1a PORTAbits.RA0 //1. LED für Raum 1 liegt an Port A RA0
#define RAUM_1b PORTAbits.RA1 //2. LED für Raum 1 liegt an Port A RA1
#define RAUM_2a PORTCbits.RC1 //1. LED für Raum 2 liegt an Port C RC1
#define RAUM_2b PORTCbits.RC2 //2. LED für Raum 2 liegt an Port C RC2
#define RAUM_3a PORTCbits.RC3 //1. LED für Raum 3 liegt an Port C RC3
#define RAUM_3b PORTCbits.RC4 //2. LED für Raum 3 liegt an Port C RC4
#define DGa PORTAbits.RA3 //1. LED für DG liegt an Port A RA3
#define DGb PORTCbits.RC5 //2. LED für DG liegt an Port C RC5

//Status der Beleuchtung der Zimmer
#define EINGANG 0
#define EINGANG_KUECHE 1
#define EINGANG_KUECHE_AUSGANG 2
#define KUECHE_AUSGANG 3
#define KUECHE 4
#define KUECHE_DG 5
#define DG_ABEND 6
#define NACHT 7
#define DG_MORGEN 8
#define KUECHE_DG_MORGEN 9
#define KUECHE_MORGEN 10
#define KUECHE_EINGANG_MORGEN 11
#define KUECHE_EINGANG_AUSGANG_MORGEN 12
#define KUECHE_EINGANG_ZWEI 13
#define EINGANG_MORGEN 14
#define ARBEIT 15

/*PROTOTYPES*/

void allLEDOff( void );
void initTimer0( void );
void restartTimer0( void );
void wait( int seconds );

/*HAUPTPROGRAMM*/

void main(void) //hier fängt das Programm an
{
    char state=0;

    //PORTA
    PORTA = 0x00;
    TRISA = 0x00;

    //PORTC
    PORTC = 0x00;
    TRISC = 0x00;

```

```

allLEDOff();           //alle LED's ausschalten
    initTimer0();      //initialisiert den Timer
    state = EINGANG;

    while(1) {         //Hauptschleife
switch( state ) {

case EINGANG:
    RAUM_1a = 1;
    RAUM_1b = 1;
    RAUM_2a = 0;
    RAUM_2b = 0;
    RAUM_3a = 0;
    RAUM_3b = 0;
    DGa = 0;
    DGb = 0;
    wait( 5 );
    state = EINGANG_KUECHE; //nächster Zustand ist Eingangszimmer und Küche
    break;

case EINGANG_KUECHE:
    RAUM_1a = 1;
    RAUM_1b = 1;
    RAUM_2a = 1;
    RAUM_2b = 1;
    RAUM_3a = 0;
    RAUM_3b = 0;
    DGa = 0;
    DGb = 0;
    wait( 5 );
    state = EINGANG_KUECHE_AUSGANG; //nächster Zustand ist ALLE Zimmer im EG
    break;

case EINGANG_KUECHE_AUSGANG:
    RAUM_1a = 1;
    RAUM_1b = 1;
    RAUM_2a = 1;
    RAUM_2b = 1;
    RAUM_3a = 1;
    RAUM_3b = 1;
    DGa = 0;
    DGb = 0;
    wait( 5 );
    state = KUECHE_AUSGANG; //nächster Zustand ist Küche und Ausgang
    break;

case KUECHE_AUSGANG:
    RAUM_1a = 0;
    RAUM_1b = 0;
    RAUM_2a = 1;
    RAUM_2b = 1;
    RAUM_3a = 1;

```

```

    RAUM_3b = 1;
    DGa = 0;
    DGb = 0;
    wait( 5 );
    state = KUECHE;
    break;
//nächster Zustand ist Küche

case KUECHE:
    RAUM_1a = 0;
    RAUM_1b = 0;
    RAUM_2a = 1;
    RAUM_2b = 1;
    RAUM_3a = 0;
    RAUM_3b = 0;
    DGa = 0;
    DGb = 0;
    wait( 5 );
    state = KUECHE_DG;
    break;
//nächster Zustand ist Küche und DG

case KUECHE_DG:
    RAUM_1a = 0;
    RAUM_1b = 0;
    RAUM_2a = 1;
    RAUM_2b = 1;
    RAUM_3a = 0;
    RAUM_3b = 0;
    DGa = 1;
    DGb = 1;
    wait( 5 );
    state = DG_ABEND;
    break;
//nächster Zustand ist DG abends

case DG_ABEND:
    RAUM_1a = 0;
    RAUM_1b = 0;
    RAUM_2a = 0;
    RAUM_2b = 0;
    RAUM_3a = 0;
    RAUM_3b = 0;
    DGa = 1;
    DGb = 1;
    wait( 10 );
    state = NACHT;
    break;
//nächster Zustand ist Nacht - alle schlafen

case NACHT:
    RAUM_1a = 0;
    RAUM_1b = 0;
    RAUM_2a = 0;
    RAUM_2b = 0;
    RAUM_3a = 0;
    RAUM_3b = 0;

```

```
DGa = 0;
DGb = 0;
wait( 20 );
state = DG_MORGEN;           //nächster Zustand ist DG Morgens
break;
```

```
case DG_MORGEN:
```

```
    RAUM_1a = 0;
    RAUM_1b = 0;
    RAUM_2a = 0;
    RAUM_2b = 0;
    RAUM_3a = 0;
    RAUM_3b = 0;
    DGa = 1;
    DGb = 1;
    wait( 7 );
    state = KUECHE_DG_MORGEN; //nächster Zustand ist DG + Küche Morgens
    break;
```

```
case KUECHE_DG_MORGEN:
```

```
    RAUM_1a = 0;
    RAUM_1b = 0;
    RAUM_2a = 1;
    RAUM_2b = 1;
    RAUM_3a = 0;
    RAUM_3b = 0;
    DGa = 1;
    DGb = 1;
    wait( 3 );
    state = KUECHE_MORGEN;   //nächster Zustand ist nur Küche Morgens
    break;
```

```
case KUECHE_MORGEN:
```

```
    RAUM_1a = 0;
    RAUM_1b = 0;
    RAUM_2a = 1;
    RAUM_2b = 1;
    RAUM_3a = 0;
    RAUM_3b = 0;
    DGa = 0;
    DGb = 0;
    wait( 7 );
    state = KUECHE_EINGANG_MORGEN; //nächster Zustand ist Küche und Eingang Morgens
    break;
```

```
case KUECHE_EINGANG_MORGEN:
```

```
    RAUM_1a = 1;
    RAUM_1b = 1;
    RAUM_2a = 1;
    RAUM_2b = 1;
    RAUM_3a = 0;
    RAUM_3b = 0;
    DGa = 0;
```

```
DGb = 0;
wait( 5 );
state = KUECHE_EINGANG_AUSGANG_MORGEN; //nächster Zustand ist alle EG Morgens
break;
```

```
case KUECHE_EINGANG_AUSGANG_MORGEN:
```

```
    RAUM_1a = 1;
    RAUM_1b = 1;
    RAUM_2a = 1;
    RAUM_2b = 1;
    RAUM_3a = 1;
    RAUM_3b = 1;
    DGa = 0;
    DGb = 0;
    wait( 3 );
    state = KUECHE_EINGANG_ZWEI; //nächster Zustand ist Küche und Eingang Zwei
    break;
```

```
case KUECHE_EINGANG_ZWEI:
```

```
    RAUM_1a = 1;
    RAUM_1b = 1;
    RAUM_2a = 1;
    RAUM_2b = 1;
    RAUM_3a = 0;
    RAUM_3b = 0;
    DGa = 0;
    DGb = 0;
    wait( 5 );
    state = EINGANG_MORGEN; //nächster Zustand ist Eingangszimmer Morgens
    break;
```

```
case EINGANG_MORGEN:
```

```
    RAUM_1a = 1;
    RAUM_1b = 1;
    RAUM_2a = 0;
    RAUM_2b = 0;
    RAUM_3a = 0;
    RAUM_3b = 0;
    DGa = 0;
    DGb = 0;
    wait( 3 );
    state = ARBEIT; //nächster Zustand ist Arbeit
    break;
```

```
case ARBEIT:
```

```
    RAUM_1a = 0;
    RAUM_1b = 0;
    RAUM_2a = 0;
    RAUM_2b = 0;
    RAUM_3a = 0;
    RAUM_3b = 0;
    DGa = 0;
```

```

    DGb = 0;
    wait( 30 );
    state = EINGANG;           //nächster Zustand ist Eingangszimmer
    break;

default:
    RAUM_1a = 1;               //sollte nie vorkommen
    RAUM_1b = 1;
    RAUM_2a = 0;
    RAUM_2b = 0;
    RAUM_3a = 0;
    RAUM_3b = 0;
    DGa = 0;
    DGb = 0;
    state = EINGANG;
    break;
}
}

void allLEDOff( void ) {
    PORTA = 0x00;              //LED's an Port A ausschalten
    PORTC = 0x00;              //LED's an Port C ausschalten
}

//Initialisiert den Timer 0 für einen Sekundentakt
void initTimer0( void ) {
    //Timer 0 ist ein 16-bit-Timer
    //Prescaler = 1:256

    OPTION_REG = 0b00000111;
    TMR1H = 0x00;
    TMR1L = 0x00;
    INTCONbits.GIE = 0;       //Interruptflag zurücksetzen
    return;
}

//Startet den Timer neu
void restartTimer0( void ) {
    int timerValue=0;

    //mit einem Prescaler von 1:256 entsteht nach einer Sekunde
    //ein Überlauf (4 MHz / 256 = 15625 Hz)
    INTCONbits.GIE = 0;       //Interruptflag zurücksetzen
    //Timer Wert für 1s = 65535 - 15625 = 49910 = 0xC2F6
    TMR1H = 0xC2;              //Das Highbyte muss zuerst
    //geschrieben werden.
    TMR1L = 0xF6;              //Wenn das Lowbyte geschrieben wird,
    //wird das Highbyte aktualisiert.
    //Dadurch wird es ermöglicht alle
    //16 bit auf einmal zu beschreiben
    //TMRO = 0xC2F6; funktioniert an
    //Stelle nicht, da zuerst das
    //Lowbyte und dann das Highbyte
    //geschrieben wird.
}

```

```
    INTCONbits.GIE = 1;          //Timer starten
    return;
}

//wartet für die angegebene Anzahl von Sekunden
void wait( int seconds ) {
    int i=0;

    //prüft, ob der Timer übergelaufen ist
    for( i=0; i<seconds; i++ ) {
        restartTimer0();
        //warten, bis der Timer übergelaufen ist
        while( INTCONbits.GIE == 0 );
    }
    return;
}
```