```c
/*
 * X10Sender.c
 *
 * Created: 06-05-2013 17:23:38
 *  Author: David
 */

#include <avr/io.h>
#include <avr/interrupt.h>
#define F_CPU 3686400UL
#include <util/delay.h>
#include "Timers.h"
#include "X10Sender.h"

/**********************************************************************
   Definerer X10 Beskeder
**********************************************************************/

#define HOUSEA  "01101001"
#define UNIT1   "01101001"
#define UNIT2   "10101001"
#define UNIT3   "01011001"
#define ON      "01011001"
#define OFF     "01011010"

#define START           "1110"
#define SUFFIX_ADR        "01"  // Unit Adress
#define SUFFIX_CMD        "10"  // Command Adress
#define WAIT          "000000"  // Six Zero-Crossings

#define SIZEOF_HOUSECODE    1
#define SIZEOF_UNITCODE     3
#define SIZEOF_FUNCTIONCODE 2

/**********************************************************************
   Initialiserer X10Flags
**********************************************************************/

unsigned short int ZeroCrossFlag = 0;


/**********************************************************************
   Initialiserer arrays/pointers
**********************************************************************/

char *HouseCodeArray_[SIZEOF_HOUSECODE] = { HOUSEA };
char *UnitCodeArray_[SIZEOF_UNITCODE] = { UNIT1, UNIT2, UNIT3 };
char *FunctionCodeArray_[SIZEOF_FUNCTIONCODE] = { ON, OFF };

char *HouseCodePtr_;
char *UnitCodePtr_;
char *FunctionCodePtr_;

/**********************************************************************
  Implementerer X10 Prototyper
**********************************************************************/

void InitX10Transmitter()
{
    // Konfigurer PD2 som indgang
    DDRD &= 0b11111011;
    // Trig på for-og bagkant.
    MCUCR |= 0b00000001;

    // Enable ekstern interrupt bit INT0
    GICR |= 0b01000000;

    // No ZeroCrossing => DDRBn = 0x00
    // Yes ZeroCrossing => DDRBn = 0xFF
    ZeroCrossFlag = 0;
    DDRB = 0x00;
}
```

```
void TransmitX10Bit( char bit )
{

    //Reset ZeroCrossing
    ZeroCrossFlag = 0;
    DDRB = 0x00;

    // Vent på interrupt, zeroX
    while( PINB == 0xFF )
    {
    }

    if ( bit == '1' )
    {
        // Genererer 120 KHz "burst" på PD5
        OCR1A = 14;
        // Længden af ét 120 KHz burst
        _1MSDelay();
        // Toggle OCR1A on compare match
        TCCR1A |= 0b01000000;
    }
    else if ( bit == '0')
    {
        // "Sluk" for 120 KHz burst på PD5
        OCR1A = 0;
        TCCR1A &= 0b10111111;
    }

    TCCR1A  |= 0b00000000;
    // Reset ZeroCrossing
    ZeroCrossFlag = 0;
    DDRB = 0x00;
}

void TransmitX10Kodestump( char *kodeStump )
{
    while (*kodeStump )
    {
        TransmitX10Bit( *kodeStump );
        kodeStump++;
    }
}

void TransmitX10Frame( int _HouseCode, int _UnitCode, int _FunctionCode )
{
    HouseCodePtr_ = HouseCodeArray_[ _HouseCode ];
    UnitCodePtr_ = UnitCodeArray_[ _UnitCode ];
    FunctionCodePtr_ = FunctionCodeArray_[ _FunctionCode ];


    // Send Frame - 11 Cycler:
    // Send adressen første gang
    TransmitX10Kodestump(START);
    TransmitX10Kodestump(HouseCodePtr_);
    TransmitX10Kodestump(UnitCodePtr_);
    TransmitX10Kodestump(SUFFIX_ADR);
    // Send adressen anden gang
    TransmitX10Kodestump(START);
    TransmitX10Kodestump(HouseCodePtr_);
    TransmitX10Kodestump(UnitCodePtr_);
    TransmitX10Kodestump(SUFFIX_ADR);

    // vent 3 cycler( 6 zerocrossing )
    TransmitX10Kodestump(WAIT);

    // Send kommando første gang
    TransmitX10Kodestump(START);
    TransmitX10Kodestump(HouseCodePtr_);
    TransmitX10Kodestump(FunctionCodePtr_);
    TransmitX10Kodestump(SUFFIX_CMD);
    // Send kommando anden gang
    TransmitX10Kodestump(START);
    TransmitX10Kodestump(HouseCodePtr_);
    TransmitX10Kodestump(FunctionCodePtr_);
    TransmitX10Kodestump(SUFFIX_CMD);

    // vent 3 cycler( 6 zerocrossing ) for næste frame
    TransmitX10Kodestump(WAIT);
}

// Zero cross interrupt
ISR (INT0_vect)
{
    ZeroCrossFlag = 1;
    DDRB = 0xFF;
}
```