```
/*****************************************************************************
 Description
   This module is a timer service to check the alignment of the IR beacon to the
   opponent's power plant.

 Notes
   Disable interrupt _S12_C5I when not checking for alignment!


 Author
   Jina Wang 02/15/2013

*****************************************************************************/

/*----------------------------- Include Files -----------------------------*/
/* include header files for this state machine as well as any machines at the
   next lower level in the hierarchy that are sub-machines to this machine*/


#include <stdio.h>
#include <stdlib.h>
#include <mc9s12e128.h>
#include <S12e128bits.h>
#include <Bin_Const.h>
#include <termio.h>
#include <hidef.h>
#include "S12eVec.h"

#include "E128_PWM.h"         //has all prescale definitions
#include "E128_SPI.h"
#include "E128_Servo.h"
#include "FAC_FSM.h"
#include "NavigationFSM.h"
#include "AlignPPService.h"
#include "DriveTrainService.h"
#include "ArtilleryFSM.h"

/*----------------------------- Module Defines ----------------------------*/
#define RED      0
#define BLUE     1
#define B_PERIOD 10500     //divide by 32 timer, 14 mS period for Blue Powerplant
#define R_PERIOD 15000     //divide by 32 timer, 20 mS period for Red Powerplant
#define numPulses 1         //number of edges to be captured

/*---------------------------- Module Variables ---------------------------*/
//Static Variables
static unsigned int uPeriod;
static unsigned int TargetTime;

/*---------------------------- Module Functions ---------------------------*/
//Private functions
static void EvalAlignment(unsigned int);
void interrupt _Vec_tim1ch5 PPAlignResponse(void);



/*------------------------------ Module Code ------------------------------*/

void InitIRDetectHardware(void)
//Set up IC5 to detect alignment with enemy's power plant
{
    //Enable timer system 1 and prescale values
    TIM1_TSCR1 = _S12_TEN;     //enable timer system
    TIM1_TSCR2 = PSCALE32;     //prescale to /32, 1 tick = 1.33 uS

    //no TIOS because using input capture
    TIM1_TCTL3 |= _S12_EDG5A;  //set IC5 to capture on rising
```

```
    TIM1_TFLG1 = _S12_C5F;      //clear any existing flag coming out of reset
    //TIM1_TIE |= _S12_C5I;       //enable local interrupt
    EnableInterrupts;           //globally enable interrupts

    //Set toggle line to show alignment has occurred
    DDRT |= ALIGNPP_DIR;        //set as output
    ALIGNPP_PORT = LO;          //set port initially LO

    //Set up target time to determine alignment for RED/BLUE
    if (QueryColor() == RED) //QueryColor returns SelfColor
        {
            TargetTime = numPulses * B_PERIOD; //if Self = red, target Blue power plant
            //printf("Target Time is: %u \r\n", TargetTime);
        }
    else
        {
            TargetTime = numPulses * R_PERIOD;
            //printf("Target Time is: %u \r\n", TargetTime);
        }
}

//Interrupt Response
void interrupt _Vec_tim1ch5 PPAlignResponse(void)
{
    static unsigned int uLastEdge = 0;
    static unsigned int count = 0;
    static unsigned int ElapsedTime = 0; //0-65355

    TIM1_TFLG1 = _S12_C5F; //clear flag

    uPeriod = TIM1_TC5 - uLastEdge;
    uLastEdge = TIM1_TC5;

    count ++;
    ElapsedTime += uPeriod; //accumulate elapsed time

    if (count == numPulses)
        {
            //call function to determine if last #edges captured matches target time
            EvalAlignment(ElapsedTime);
            count = 0; //reset count
            ElapsedTime = 0; //reset elapsed time
        }
}

//private function to evaluate if aligment has been achieved
static void EvalAlignment(unsigned int ElapsedTime_)
{
    if ((ElapsedTime_ < (TargetTime + 25)) && (ElapsedTime_ > (TargetTime - 25)))
        {
            ///pull line high to light LED
            ALIGNPP_PORT = HI;
        }
    return;
}


/*******************PUBLIC FUNCTIONS****************/
void EnableIRDetection ( boolean input_)
{
    if (input_ == True)
        {
            TIM1_TIE |= _S12_C5I;
        }
    else
        {
            TIM1_TIE &= BIT5LO;
```

```
        }
}
```