```
/*******************************************************************************
 Module
   DriveTrainService.c

 Revision
   1.0.0

 Description
   This service uses functions from the PWM module to implement commands for
   robot drive. Public functions from this module will be used by state machines
   to drive the motor in forward, reverse, turning operations, and other drive
   instructions required to meet the objectives of Lab 8.

 Notes

 History
 When           Who     What/Why
 -------------- ---     --------
 02/02/2013     BMJ     Started Coding
*******************************************************************************/
/*----------------------------- 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 "S12eVec.h"

#include "ES_Configure.h"
#include "ES_Framework.h"
#include "DriveTrainService.h"
#include "E128_PWM.h"
#include "NavigationFSM.h"
#include "AlignPPService.h"
#include "ArtilleryFSM.h"
#include "StrategyFSM.h"
#include "FAC_FSM.h"

/*----------------------------- Module Defines ----------------------------*/

#define ControlConstant 1;

/*---------------------------- Module Functions ---------------------------*/
/* prototypes for private functions for this service.They should be functions
   relevant to the behavior of this service
*/

void interrupt _Vec_tim0ch4 PropMotorControl(void);

/*---------------------------- Module Variables ---------------------------*/
// with the introduction of Gen2, we need a module level Priority variable
static uint8_t MyPriority;
static signed int deltaPWM;
static signed char Direction;


/*------------------------------ Module Code ------------------------------*/
/*******************************************************************************
 Function
     InitDriveTrainService

 Parameters
     uint8_t : the priorty of this service
```

```
 Returns
     boolean, False if error in initialization, True otherwise

 Description
     Saves away the priority, and does any
     other required initialization for this service
 Notes

 Author
     Jina Wang, 01/16/12, 10:00
****************************************************************************/
boolean InitDriveTrainService ( uint8_t Priority )
{
    ES_Event ThisEvent;

    MyPriority = Priority;

    // post the initial transition event
    ThisEvent.EventType = ES_INIT;
    if (ES_PostToService( MyPriority, ThisEvent) == True)
        {
            return True;
        }
    else
        {
            return False;
        }
}

/****************************************************************************
 Function
     PostDriveTrainService

 Parameters
     EF_Event ThisEvent ,the event to post to the queue

 Returns
     boolean False if the Enqueue operation failed, True otherwise

 Description
     Posts an event to this state machine's queue
 Notes

 Author
     Jina Wang, 10/23/11, 19:25
****************************************************************************/
boolean PostDriveTrainService( ES_Event ThisEvent )
{
    return ES_PostToService( MyPriority, ThisEvent);
}

/****************************************************************************
 Function
    RunDriveTrainService

 Parameters
   ES_Event : the event to process

 Returns
   ES_Event, ES_NO_EVENT if no error ES_ERROR otherwise

 Description
   add your description here
 Notes

 Author
```

```
   Jina Wang, 01/15/12, 15:23
*****************************************************************************/
ES_Event RunDriveTrainService( ES_Event ThisEvent )
{
    ES_Event ReturnEvent;
    ReturnEvent.EventType = ES_NO_EVENT; // assume no errors
    /****************************************************************************

                           BEGIN STATE MACHINE CODE

    ****************************************************************************/
    switch (ThisEvent.EventType)
        {
        case DRIVE:
            switch (ThisEvent.EventParam)
                {
                case FORWARD:
                    DriveForwardFull(0);
                    Direction = FORWARD;
                    TurnOnPControl();  //Enable P-Control Interrupt
                    break; //end DriveForwardFull

                case FORWARD_HALF:
                    DriveForwardHalf();
                    TurnOffPControl();   //Disable P-Control Interrupt
                    break;

                case REVERSE:
                    DriveReverseFull(0);
                    Direction = REVERSE;
                    TurnOnPControl();  //Enable P-Control Interrupt
                    break; //end DriveReverseFull

                case REVERSE_HALF:
                    DriveReverseHalf();
                    TurnOffPControl();   //Disable P-Control Interrupt
                    break;

                }
            break; // end Drive case

        case ROTATE:
            switch (ThisEvent.EventParam)
                {
                case CW:
                    RotateClockwise();
                    TurnOffPControl();   //Disable P-Control Interrupt
                    break; //end RotateClockwise

                case CCW:
                    RotateCounterClockwise();
                    TurnOffPControl();   //Disable P-Control Interrupt
                    break; //end RotateCounterClockwise
                }
            break; //end Rotate case

        case ROTATE_HALF:
            switch (ThisEvent.EventParam)
                {
                case CW:
                    RotateClockwiseHalf();
                    TurnOffPControl();   //Disable P-Control Interrupt
                    break;

                case CCW:
                    RotateCounterClockwiseHalf();
                    TurnOffPControl();   //Disable P-Control Interrupt
```

```
                    break;

                }
            break;


        case STOP_MOTOR:
            // Received Stop Motor Event
            StopMotor();
            TurnOffPControl();   //Disable P-Control Interrupt
            break; //end Rotate case

        case ALIGNPP:
            RotateClockwise();
            TurnOffPControl();   //Disable P-Control Interrupt
            break;

        case BACKUP_HALF:
            DriveReverseHalf();
            TurnOffPControl();   //Disable P-Control Interrupt
            break;

        }
    return ReturnEvent;
}

/*****************************************************************************
 Functions
    Enable/Disable Timer0Ch4 Interrupt for P-Control

 Parameters
   None

 Returns
   Nothing

 Description
   When called, these functions enable/disable the TIM0_Ch4 interrupt

 Notes

 Author
   Ben Sagan 03/05/2013
*****************************************************************************/
void TurnOnPControl( void )
{
    TIM0_TIE |= _S12_C4I;  //Enable P-Control Interrupt
}

void TurnOffPControl( void )
{
    TIM0_TIE &= ~_S12_C4I;   //Disable P-Control Interrupt
}

/*****************************************************************************
 Function
    Proportional Motor Control

 Parameters
   None

 Returns
   Nothing

 Description
   When called, this function posts to E128_PWM with control
```

```
 Notes

 Author
   Ben Sagan 03/04/2013
*****************************************************************************/
void interrupt _Vec_tim0ch4 PropMotorControl(void)
{
    unsigned char DesiredTheta;
    unsigned char CurTheta;
    signed int Error;

    DesiredTheta = QueryDesiredTheta();
    CurTheta = QueryTheta(SelfNum);

    Error = DesiredTheta - CurTheta;

    deltaPWM = Error * ControlConstant;

    if (CurTheta != 0)
        {
            if (Direction == FORWARD)
                {
                    DriveForwardFull( deltaPWM );
                }
            else if (Direction == REVERSE)
                {
                    DriveReverseFull( deltaPWM );
                }
        }

    TIM0_TFLG1 = _S12_C4F;
    TIM0_TC4 += 25000;//18750; // 100 ms

}
```