```
/*******************************************************************************
 Module
   NeedleService.c

 Revision
   1.0

 Description
   This is a file for implementing the moving needle (period of time) service

 Notes

 History
 When           Who     What/Why
 -------------- ---     --------

*******************************************************************************/
/*---------------------------- 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 <stdint.h>
#include <stdbool.h>
#include "inc/hw_gpio.h"
#include "inc/hw_types.h"
#include "inc/hw_pwm.h"
#include "inc/hw_memmap.h"
#include "driverlib/sysctl.h"

#include "ES_Configure.h"
#include "ES_Framework.h"
#include "NeedleService.h"
#include "GameService.h"
#include "PWM16Tiva.h"


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

/*---------------------------- Module Functions ----------------------------*/
/* prototypes for private functions for this service.They should be functions
   relevant to the behavior of this service
*/
static void UpdateNeedle (void);
static void MoveNeedleRight (void);
static void MoveNeedleLeft (void);
/*---------------------------- Module Variables ----------------------------*/
// with the introduction of Gen2, we need a module level Priority variable
static uint8_t MyPriority;

static NeedleServiceState_t CurrentState = NeedleLeft;

static uint16_t StepSize = 25;
static const uint16_t channel = 2;
static uint16_t CurrentPulse = 3000;
static const uint16_t NeedleTime = 2000; //ms
static const uint16_t reqPeriod = 25000;
static uint8_t group = 1;



/*---------------------------- Module Code ----------------------------*/
/*******************************************************************************
 Function
     InitNeedleService
```

```
 Parameters
     uint8_t : the priorty of this service

 Returns
     bool, false if error in initialization, true otherwise

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

 Author
     Kat Liu
****************************************************************************/
bool InitNeedleService(uint8_t Priority)
{
  ES_Event_t ThisEvent;
  MyPriority = Priority;

  //static uint8_t PWM_NumChannels = 5;

  //(PWM_NumChannels);
  PWM_TIVA_SetPeriod(reqPeriod, group);

    if (ES_PostToService( MyPriority, ThisEvent) == true)
  {
      return true;
  }else
  {
      return false;
  }
}
/****************************************************************************
 Function
     PostNeedleService

 Parameters
     EF_Event ThisEvent ,the event to post to the queue

 Returns
     bool false if the Enqueue operation failed, true otherwise

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

 Author
     Kat Liu
****************************************************************************/
bool PostNeedleService(ES_Event_t ThisEvent)
{
  return ES_PostToService(MyPriority, ThisEvent);
}

/****************************************************************************
 Function
    RunNeedleService

 Parameters
   ES_Event : NEEDLE_START, CELEB_TIMEOUT, GAME_START

 Returns
   ES_Event, ES_NO_EVENT
```

```
 Description
   Implements needle service to move needle from left to right
 Notes

 Author
   Kat Liu
****************************************************************************/
ES_Event_t RunNeedleService(ES_Event_t ThisEvent)
{
  ES_Event_t ReturnEvent;
  ReturnEvent.EventType = ES_NO_EVENT; // assume no errors
  NeedleServiceState_t NextState;
  NextState = CurrentState;


  switch(CurrentState){

    //CurrentState is NeedleLeft
    case NeedleLeft:

      printf("NeedleLeft\r\n");
      //MoveNeedleLeft();


      if (ThisEvent.EventType == NEEDLE_START) {
        //Start needle timer
        ES_Timer_InitTimer(NEEDLE_TIMER, NeedleTime);
        NextState = NeedleMoving;
        printf("NEEDLE TIMER STARTED!!!!!!!!!\r\n");
      }
      break;

    //Current State is NeedleMoving
    case NeedleMoving:

      printf("NeedleMoving\r\n");

      if (ThisEvent.EventType == ES_TIMEOUT && ThisEvent.EventParam ==
NEEDLE_TIMER) {
        //Call UpdateNeedle function
        UpdateNeedle();
        printf("NEEDLE TIMEOUT\r\n");
        ES_Timer_InitTimer(NEEDLE_TIMER, NeedleTime);
      }
      else if (ThisEvent.EventType == ES_TIMEOUT && ThisEvent.EventParam ==
CELEB_TIMER) {
        //Move needle to the right most position
        MoveNeedleLeft();
        NextState = NeedleLeft;
        printf("Move Leftt\r\n");
      }
      break;

    //Current State is NeedleRight
//    case NeedleRight :
//
//      printf("Needle Right\r\n");
//
//      if (ThisEvent.EventType == NEEDLE_START) {
//        //Move needle to the left most position
//        MoveNeedleLeft();
//        NextState = NeedleLeft;
//        printf("Left");
```

```
//      }
//      break;
    }

    //Set CurrentState to NextState
    CurrentState = NextState;
    return ReturnEvent;
}

/*****************************************************************************
 private functions
 *****************************************************************************/
/*****************************************************************************
 Function
   UpdateNeedle

 Parameters
   None

 Returns
   None

 Description
   Actuates motor to move the needle incrementally

 Notes

 Author
   Katherine Liu, 11/11/19
*****************************************************************************/
static void UpdateNeedle (void)
{
  CurrentPulse -= StepSize;
  if (CurrentPulse > 2900) { CurrentPulse = 2900;}
  if (CurrentPulse < 1225) { CurrentPulse = 1250;}
  PWM_TIVA_SetPulseWidth(CurrentPulse, channel);
  printf("Needle Pos: %d\r\n", CurrentPulse);
  }

  /*****************************************************************************
 Function
   MoveNeedleRight

 Parameters
   None

 Returns
   None

 Description
   Actuates motor to move the needle to the most maximum right position

 Notes

 Author
   Katherine Liu, 11/11/19
*****************************************************************************/
static void MoveNeedleRight (void)
{
  static uint16_t PulseWidth = 3000;
  PWM_TIVA_SetPulseWidth(PulseWidth, channel);
  //PWM_TIVA_SetPeriod(reqPeriod, group);
  CurrentPulse = PulseWidth;
  printf("Needle Right: %d\r\n", CurrentPulse);
```

```
  }
/*****************************************************************************
 Function
   MoveNeedleLeft

 Parameters
   None

 Returns
   None

 Description
   Actuates motor to move the needle to the most maximum left position

 Notes

 Author
   Katherine Liu, 11/11/19
*****************************************************************************/
static void MoveNeedleLeft (void)
{
    static uint16_t PulseWidth = 2900;
    PWM_TIVA_SetPulseWidth(PulseWidth, channel);
    //PWM_TIVA_SetPeriod(reqPeriod, group);
    CurrentPulse = PulseWidth;
    printf("Needle Left: %d\r\n", CurrentPulse);
  }

/*------------------------------- Footnotes -------------------------------*/
/*------------------------------ End of file ------------------------------*/
```