```
/*******************************************************************************
 Module
   DAC_Service.c

 Revision
   1.0.1

 Description
   This implements a Digital-to-Analog output service to generate audio tones

 Notes

 History
 When           Who     What/Why
 -------------- ---     --------
 05/14/20 22:05 hbf     began conversion for final project
 05/06/20 08:22 hbf     began conversion for Lab 10
 01/16/12 09:58 jec     began conversion from TemplateFSM.c
*******************************************************************************/
/*---------------------------- 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 "ES_Configure.h"
#include "ES_Framework.h"
#include "ES_Port.h"
#include "DAC_Service.h"
#include "dbprintf.h"

/*---------------------------- Module Functions ----------------------------*/
/* prototypes for private functions for this service.They should be functions
   relevant to the behavior of this service
*/
static void InitTimer0(void);
static void InitDAC(void);
/*---------------------------- Module Variables ----------------------------*/
// with the introduction of Gen2, we need a module level Priority variable
static uint8_t MyPriority;
const static uint8_t sin[] = {16, 19, 22, 25, 27, 29, 30, 31, 31, 31, 29, 28, 26, 23, 20, 18, 14, 12, 9, 6, 4, 3, 1, 1, 1, 2, 3, 5, 7, 10, 13};
const static uint8_t sin_length = 31;
static int sin_index = 0;

static uint8_t CMP_VAL[4] = {129, 103, 77, 51};
/*---------------------------- Module Code ----------------------------*/
/*******************************************************************************
 Function
    InitDACService

 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
     H. Francis, 05/06/20, 10:00
*******************************************************************************/
bool InitDACService(uint8_t Priority)
{
  ES_Event_t ThisEvent;
```

```
  MyPriority = Priority;

  InitTimer0();
  InitDAC();

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

/****************************************************************************
 Function
   PostDACService

 Parameters
    EF_Event_t 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
    H. Francis, 05/06/20, 10:38
****************************************************************************/
bool PostDACService(ES_Event_t ThisEvent)
{
  return ES_PostToService(MyPriority, ThisEvent);
}

/****************************************************************************
 Function
   RunDACService

 Parameters
   ES_Event_t : the event to process

 Returns
   ES_Event, ES_NO_EVENT if no error ES_ERROR otherwise

 Description
   no action needed
 Notes

 Author
   H. Francis, 05/06/20, 10:40
****************************************************************************/
ES_Event_t RunDACService(ES_Event_t ThisEvent)
{
  ES_Event_t ReturnEvent;
  ReturnEvent.EventType = ES_NO_EVENT; // assume no errors
  if(ThisEvent.EventType == ES_INIT)
  {
  }
  return ReturnEvent;
}

/****************************************************************************
 Function
```

```
   getNextDAC

 Parameters
   none

 Returns
   uint8_t: value to program DAC

 Description
   getter function that tracks a step function along a sine wave
 Notes

 Author
   H. Francis, 05/06/20, 11:00
*******************************************************************************/
uint8_t getNextDAC(void)
{
   sin_index++;
   if(sin_index == sin_length) {
      sin_index = 0;
   }
   return sin[sin_index];
}
/*******************************************************************************
 Function
   EnableDAC

 Parameters
   none

 Returns
   none

 Description
   enables DAC service when preparing for a transmission
 Notes

 Author
   H. Francis, 05/06/20, 11:08
*******************************************************************************/
void EnableDAC(void)
{
   DAC1CON1 = getNextDAC();
   TMR0IF = 0;
   TMR0IE = 1;
   T0EN = 1;
   DAC1EN = 1;
}
/*******************************************************************************
 Function
   SendDibit

 Parameters
   none

 Returns
   none

 Description
   sets the timer compare value to send a tone
 Notes

 Author
   H. Francis, 05/14/20, 22:15
*******************************************************************************/
void SendDibit(uint8_t value)
{
   T0EN = 0;
```

```
    TMR0H = CMP_VAL[value];
    TMR0L = 0;
    TMR0IF = 0;
    TMR0IE = 1;
    T0EN = 1;
}
/*****************************************************************************
 Function
    DisableDAC

 Parameters
    none

 Returns
    none

 Description
    disables DAC service when finished with a transmission
 Notes

 Author
   H. Francis, 05/06/20, 11:09
*****************************************************************************/
void DisableDAC(void)
{
    DAC1CON1 = 0;
    DAC1EN = 0;
    TMR0IE = 0;
    T0EN = 0;
}
/*****************************************************************************
 private functions
 *****************************************************************************/
/*****************************************************************************
 Function
    InitTimer0

 Parameters
    none

 Returns
    none

 Description
    initializes Timer 0 to be used with DAC output of a certain frequency
 Notes

 Author
   H. Francis, 05/06/20, 11:09
*****************************************************************************/
static void InitTimer0(void)
{
    //Set to 8-bit time
    //Prescale 1:2 T0Con1 0001
    T016BIT = 0;
    T0CS1 = 1;
    T0CKPS0 = 1;
    TMR0H = CMP_VAL[0]; //start LO
}
/*****************************************************************************
 Function
    InitDAC

 Parameters
    none

 Returns
    none
```

```
 Description
   initializes DAC, used to output a frequency for tone generation
 Notes

 Author
   H. Francis, 05/06/20, 11:09
*******************************************************************************/
static void InitDAC(void)
{
   CDAFVR0 = 1; //use 1.024 as ref voltage
   FVREN = 1; //enable fixed voltage ref
   while(!FVRRDY) {;} //Wait for periph to be ready
   DAC1PSS1 = 1; //set positive ref voltage to be FVR
   DAC1CON1 = sin[sin_index]; //set first value
   DAC1OE2 = 1; //set output to be RB7
   DAC1EN = 1; //enable

   //Enable all interrupts
   GIE = 1;
   PEIE = 1;
}
/*------------------------------ Footnotes -------------------------------*/
/*------------------------------ End of file -------------------------------*/
```