```
/*******************************************************************************
 Module
   MicInputService.c

 Revision
   1.0.1

 Description
   This implements signal conditioning from a microphone input to parse audio tones

 Notes

 History
 When           Who     What/Why
 -------------- ---     --------
 05/17/20       LG      instituted CheckSum and updated based on state machine
 05/16/20       LG      tried out a simpler version of receive (one sample based on timeout)
 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 "MicInputService.h"
#include "dbprintf.h"
#include "RX_Service.h"

/*----------------------------- Module Defines ----------------------------*/
#define TARGET_DIBIT_0 1000 // 1000
#define TARGET_DIBIT_1 813 //1250
#define TARGET_DIBIT_2 604 //1667 Hz
#define TARGET_DIBIT_3 459 //2500
#define MAX_PERIOD_TICKS 1200
#define MIN_PERIOD_TICKS 300
#define ERROR 75
#define RESET_TIME 20
#define HALF_BIT_TIME 5
/*---------------------------- Module Functions ---------------------------*/
/* prototypes for private functions for this service.They should be functions
   relevant to the behavior of this service
*/
static void InitCCP2(void);
/*---------------------------- Module Variables ---------------------------*/
// with the introduction of Gen2, we need a module level Priority variable
static uint8_t MyPriority;
static uint16_t Period;
static uint16_t LastCapture = 0;
static uint8_t LastDibitDetected;
static bool NotListening = true;
static uint8_t LastTime = 0;
static uint8_t LastCounter;
static uint8_t Counter;
static uint8_t BadPeriodCounter;
static uint8_t LastBadPeriodCounter;
static uint8_t BadPeriods;
/*------------------------------ Module Code ------------------------------*/
/*******************************************************************************
 Function
    InitMicInputService

 Parameters
     uint8_t : the priority 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, 05/07/20, 14:00
****************************************************************************/
bool InitMicInputService(uint8_t Priority)
{
 ES_Event_t ThisEvent;

 MyPriority = Priority;

 InitCCP2();
 // post the initial transition event
 ThisEvent.EventType = ES_INIT;
 if (ES_PostToService(MyPriority, ThisEvent) == true)
 {
  return true;
 }
 else
 {
  return false;
 }
}

/****************************************************************************
 Function
   PostMicInputService

 Parameters
    ES_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, 05/07/20, 14:20
****************************************************************************/
bool PostMicInputService(ES_Event_t ThisEvent)
{
 return ES_PostToService(MyPriority, ThisEvent);
}

/****************************************************************************
 Function
   RunMicInputService

 Parameters
  ES_Event_t : the event to process

 Returns
  ES_Event, ES_NO_EVENT if no error ES_ERROR otherwise

 Description
  Runs this state machine
 *
 Notes
```

```
 Author
   H. Francis, 05/07/20, 14:20
*****************************************************************************/
ES_Event_t RunMicInputService(ES_Event_t ThisEvent)
{
  ES_Event_t ReturnEvent;
  ReturnEvent.EventType = ES_NO_EVENT; // assume no errors

    if(ThisEvent.EventType == ES_TIMEOUT && ThisEvent.EventParam == PERIOD_RESET_TIMER)
    {
       Period = 0;

    }
  return ReturnEvent;
}

/****************************************************************************
 Function
   Check4PeriodChange

 Parameters
   none

 Returns
   bool, true if a start bit has been found, false otherwise

 Description
   Event Checker that

 Notes

 Author
  L. Gardner, 5/16/2020
*****************************************************************************/
bool Check4PeriodChange(void)
{
   bool ReturnVal = false;
   ES_Event_t PostEvent;
   ++Counter;
   if(NotListening)
   {
      Period = 0;
      LastDibitDetected = NOT_VALID_DIBIT;
      return false;
   } else if (((Period > (TARGET_DIBIT_3 - ERROR)) && (Period < (TARGET_DIBIT_3 + ERROR))) && (GetRXState()==1)) {
      if(Counter == (LastCounter + 1)) ++LastTime;
      else LastTime = 0;
      if(LastTime == 4) {
         PostEvent.EventType = ES_START_BIT_DETECTED;
         PostRXService(PostEvent);
         //InStartBit = true;
         ReturnVal = true;
         ES_Timer_InitTimer(RX_TIMER, HALF_BIT_TIME);
      }
   }
   LastCounter = Counter;
   return ReturnVal;
}

/****************************************************************************
 Function
   QueryBitState

 Parameters
   none

 Returns
 The BitState
```

```
 Description
 Function that updates the last dibit detected based on whether it falls within a tolerance for the period

 Notes

 Author
 L. Gardner, 05/16/2020
****************************************************************************/
uint8_t QueryBitState(void)
{
    if (Period == 0) {
      ES_Event_t ThisEvent;
      ThisEvent.EventType = ES_BAD_RECEPTION;
      PostRXService(ThisEvent);
      printf("Bad receive abort \n\r");
 }
    else if ((Period > (TARGET_DIBIT_0 - ERROR)) && (Period < TARGET_DIBIT_0 + ERROR)) {
      LastDibitDetected = DIBIT_0;
 }

   else if ((Period > (TARGET_DIBIT_1 - ERROR)) && (Period < TARGET_DIBIT_1 + ERROR)) {
      LastDibitDetected = DIBIT_1;
 }

   else if ((Period > (TARGET_DIBIT_2 - ERROR)) && (Period < TARGET_DIBIT_2 + ERROR)) {
      LastDibitDetected = DIBIT_2;
 }

   else if ((Period > (TARGET_DIBIT_3 - ERROR)) && (Period < TARGET_DIBIT_3 + ERROR)) {
      LastDibitDetected = DIBIT_3;
 }

   return LastDibitDetected;
}
/****************************************************************************
 Function
 ClearFirstTime

 Parameters
   none

 Returns
  none

 Description
 Clears the first time that we have checked for a start bit recently

 Notes

 Author
  L. Gardner, 5/15/2020
****************************************************************************/

void ClearFirstTime(void) {
   //FirstTime = 0;
   LastTime = 0;
   return;
}

/****************************************************************************
 Function
   CCP2IntHandler

 Parameters
   none

 Returns
```

```
  none

 Description
  ISR for CCP2

 Notes

 Author
  H. Francis, 5/7/2020, 18:46
*****************************************************************************/
void CCP2IntHandler(void)
{
   uint16_t NewCapture = ((uint16_t)CCPR2H << 8) | CCPR2L;
   uint16_t NewPeriod = NewCapture - LastCapture;
   if(NewPeriod < MAX_PERIOD_TICKS && NewPeriod > MIN_PERIOD_TICKS) {
      Period = NewPeriod;
   }
   LastCapture = NewCapture;
   ES_Timer_InitTimer(PERIOD_RESET_TIMER, RESET_TIME);
}
/*****************************************************************************
 Function
   EnableListen

 Parameters
   none

 Returns
  none

 Description
  Enables the capture compare module for the receive module

 Notes

 Author
  H. Francis, 5/7/2020, 18:50
*****************************************************************************/
void EnableListen(void)
{
   NotListening = false;
   TMR1ON = 1;
   CCP2IF = 0;
   CCP2IE = 1;
   CCP2EN = 1;
}
/*****************************************************************************
 Function
   DisableListen

 Parameters
   none

 Returns
  none

 Description
  Disables the capture compare module for the receive module

 Notes

 Author
  H. Francis, 5/7/2020, 18:58
*****************************************************************************/
void DisableListen(void)
{
   NotListening = true;
   CCP2EN = 0;
```

```
    CCP2IE = 0;
    TMR1ON = 0;
}

/*******************************************************************************
 private functions
 ******************************************************************************/
/*******************************************************************************
 Function
    InitCCP2

 Parameters
    none

 Returns
   none

 Description
   Initializes the CCP function along with timer 1

 Notes

 Author
  H. Francis, 5/7/2020, 14:50
*******************************************************************************/
static void InitCCP2(void)
{
    //Set up TMR1
    //Clock select: Fosc/4: 0001
    T1CS0 = 1;
    //Prescale = 1:8
    T1CKPS1 = 1;
    T1CKPS0 = 1;
    TMR1IE = 0;
    TMR1IF = 0;

    //Set up port (RC1) as digital input
    ANSELC &= BIT1LO;
    TRISC |= BIT1HI;

    //Set up test port
    ANSELC &= BIT4LO;
    LATC &= BIT4LO; //start lo
    TRISC &= BIT4LO;

    //Set up CCP2
    //Disable interrupt
    CCP2IE = 0;
    //Mode = Capture, every rising edge, 0101
    CCP2MODE2 = 1;
    CCP2MODE0 = 1;

    //Clear Interrupt
    CCP2IF = 0;
}

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