Pseudocode for MicInputService

Define:

MAX\_PERIOD\_TICKS

MIN\_PERIOD\_TICKS

TARGET\_DIBIT\_0 1000 // 1000

TARGET\_DIBIT\_1 813 //1250

TARGET\_DIBIT\_2 604 //1667 Hz

TARGET\_DIBIT\_3 459 //2500

ERROR

RESET\_TIME

HALF\_BIT\_TIME

Module Functions:

InitCCP2

Module Variables:

MyPriority

Period

LastCapture

LastDibitDetected

NotListening (bool)

LastTime

LastCounter

Counter

BadPeriodCounter

LastBadPEriodCounter

BadPeriods

Function: InitMicInputService

Params: Priority

Returns: bool

Set MyPriority

InitCCP2

End InitMicInputService

Function: PostMicInputService

Params: ES\_Event\_t

Returns: bool

Post to this service

End PostMicInputService

Function: RunMicInputService

Params: ES\_Event\_t

Returns: ES\_Event\_t

if ThisEvent is ES\_TIMEOUT of PERIOD\_RESET\_TIMER

Set Period to 0

endif

End RunMicInputService

Function: Check4PeriodChange

Params: none

Returns: bool

Increment Counter

If not listening

Set period to 0

Set LastDibitDetected to NOT\_VALID\_DIBIT

Return false

Else if period is in range for start dibit

If counter equals lastCounter + 1, increment lastTime

Else set last time to 0

Endif

If lastTime is 4

Post StartBitDetected

Start half-bit timer

Endif

Endif

End Check4PeriodChange

Function: QueryBitState

Params: none

Returns: uint8\_t bitstate

If Period is 0

Post Bad Reception

Else if period is in Dibit 0 range

Set LastDibitDetected to Dibit 0

Else if period is in Dibit 1 range

Set LastDibitDetected to Dibit 1

Else if period is in Dibit 2 range

Set LastDibitDetected to Dibit 2

Else if period is in Dibit 3 range

Set LastDibitDetected to Dibit 3

Endif

Return LastDibitDetected

End QueryBitState

Function: ClearFirstTime

Params: none

Returns: none

Set LastTime to 0

End ClearFirstTime

Function: CCP2IntHandler

Params: none

Returns: none

Get new Capture value

Calculate new period

If New Period is less than max period ticks and greater than min period ticks,

Store new period

Endif

Set Last Capture to NewCapture

Start Period reset timer

End CCP2IntHandler

Function: EnableListen

Params: none

Returns: none

Set NotListening to false

Enable TMR1

Clear CCP2 interrupt flag

Enable CCP2 interrupt

Enable CCP2

End EnableListen

Function: DisableListen

Params: none

Returns: none

Set NotListening to true

Disable TMR1

Disable CCP2 interrupt

Disable CCP2

End DisableListen

Function: InitCCP2

Params: none

Returns: none

Select clock Fosc/4

Set Prescale to 1:8

Disable tmr1 interrupt, clear flag

Set up port (RC1) as digital input

Set up CCP2, disable interrupt

Set mode to capture, every rising edge

Clear interrupt

End InitCCP2