

Module Functions

```
// static void UpdateNeedle (void);  
// static void MoveNeedleRight (void);  
// static void MoveNeedleLeft (void);
```

Module Variables

```
// static uint8_t MyPriority;  
// static NeedleServiceState_t CurrentState = NeedleLeft;  
// static uint16_t StepSize to have noticeable change  
// static const uint16_t channel  
// static uint16_t CurrentPulse to maximum to right  
// static const uint16_t NeedleTime to 2 seconds  
// static const uint16_t reqPeriod = 25000;  
// static uint8_t group = 1;
```

Function

InitNeedleService

Parameters

uint8_t : the priority of this service

Returns

bool, false if error in initialization, true otherwise

```
// Initialize InitNeedleService and pass into Priority variable  
// Initialize the MyPriority variable with the passed in parameter  
// Set PWM channels and periods  
// end function
```

Function

PostNeedleService

Parameters

EF_Event ThisEvent ,the event to post to the queue

Returns

bool false if the Enqueue operation failed, true otherwise

```
// Initialize PostNeedleService and pass into ThisEvent  
// Post to the service this event  
// end function
```

Function

RunNeedleService

Parameters

ES_Event : NEEDLE_START, CELEB_TIMEOUT, GAME_START

Returns

ES_Event, ES_NO_EVENT

```
// Initialize RunNeedleService and pass into ThisEvent
```

```
// assume no errors
```

```
// Set Current state to next state
```

```
// switch current state
```

```
// If CurrentState is NeedleRight
// move needle right
// Start needle timer
// next state is NeedleMoving
// end if
// end case
// if Current State is NeedleMoving
// Call UpdateNeedle function
// Set NeedleTimer
// if eventType is timeout and parameter is celeb_timer
// Move needle to the left most position
// end if
// set next state to needleLeft
// end if
// end case
// Set CurrentState to NextState
// end function
```

Function

UpdateNeedle

Parameters

None

Returns

None

```
// Initialize UpdateNeedle
// Set current pulse to increment by desired StepSize
// if Current Pulse is past the most right position
// set current pulse to the most right position
// end if
// if Current Pulse is not at the most left position
// set current pulse to the most left position
// end if
// set current pulse to PWM Tiva
// end function
```

Function

MoveNeedleRight

Parameters

None

Returns

None

Description

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

```
// Initialize MoveNeedleRight
// set initial pulsewidth to the most right position
```

```
// set PulseWidth to PWM Tiva
// set current pulse to pulse width
// end function
Function
MoveNeedleLeft
Parameters
None
Returns
None
Description
Actuates motor to move the needle to the most maximum left position
// initialize MoveNeedleLeft
// set PulseWidth to most left position
// Set PulseWidth on PWM Tiva
// Set current pulse to pulse width
// end function
```