Pseudo-code for the SpotlightControl module (a service that implements a state machine)

****

Data private to the module: MyPriority, CurrentState, LastBrightState, LastDimState

Functions private to the module: InitBrightnessInputs

Event checkers: CheckBrightButton, CheckDimButton

#defines:

BrightButton_Read, DimButton_Read //Port registers to attach pins

BrightButton_TRIS, DimButton_TRIS //TRIS registers to initialize the pins as outputs

InitSpotlightControl

Takes a priority number, returns True.

Initialize brightness inputs by calling InitBrightnessInputs()

Post Event ES_Init to TargetLEDs queue (this service)

End of InitSpotlightControl

RunSpotlightControl (implements the state machine for SpotlightControl)

EventType field of ThisEvent will be one of: ES_INIT, ES_DISCO, ES_GAME_OVER, ES_NEW_TARGET, ES_REPLAY,

ES_TIMEOUT. The parameter field of the ThisEvent will be indicating one timer that has expired in case of ES_TIMEOUT.

Returns ES_NO_EVENT

Based on the state of the CurrentState variable choose one of the following blocks

of code:

CurrentState is Spotlight_INIT

Based on the event received, choose one of the following blocks of code:

EventType of ThisEvent is ES_INIT

```
        Set CurrentState to Spotlight_IDLE
End SpotlighControl_INIT block

CurrentState is Spotlight_IDLE
   Based on the event received, choose one of the following blocks of code:
      EventType of ThisEvent is ES_BRIGHTNESS
        Post ES_NEW_MESSAGE with message parameter to LED Display Service
        Set CurrentState to BRIGHTNESS_LEVEL_CONTROL

End Spotlight_IDLE block

CurrentState is BRIGHTNESS_LEVEL_CONTROL
   Based on the event received, choose one of the following blocks of code:
      EventType of ThisEvent is ES_NEW_GAME
        Post ES_NEW_MESSAGE with message parameter to LED Display Service
        Set CurrentState to READY

      EventType of ThisEvent is ES_BRIGHT_BUTTON
        Post ES_BRIGHT_BUTTON event to PWM service
        Set current state to BRIGHTNESS_LEVEL_CONTROL

      EventType of ThisEvent is ES_DIM_BUTTON
        Post ES_DIM_BUTTON event to PWM service
        Set current state to BRIGHTNESS_LEVEL_CONTROL

      EventType of ThisEvent is ES_TIMEOUT
       If ThisEvent.EventParam == IDLE_TIMER
        Set current state to SPOTLIGHT_IDLE
       End if
End BRIGHTNESS_LEVEL_CONTROL block
```

```
CurrentState is READY
   Based on the event received, choose one of the following blocks of code:
        EventType of ThisEvent is ES_JOYSTICK_LR or ES_JOYSTICK_UD
          Post ES_GAME_TONE event to Buzzer sevice
          Set CurrentState to MOVING

       EventType of ThisEvent is ES_TIMEOUT
          if EventParam of ThisEvent is IDLE_TIMER
             Set CurrentState to Spotlight_IDLE
          End If

End READY block

CurrentState is MOVING
   Based on the event received, choose one of the following blocks of code:
      EventType of ThisEvent is ES_JOYSTICK_LR
           Post ES_JOYSTICK_LR to PWMService
          Set CurrentState to MOVING

      EventType of ThisEvent is ES_JOYSTICK_UD
          Post ES_BRIGHT_BUTTON event to PWM service
           Post ES_JOYSTICK_UD to PWMService
          Set CurrentState to MOVING

      EventType of ThisEvent is ES_TIMEOUT
       If ThisEvent.EventParam == IDLE_TIMER
          Set current state to SPOTLIGHT_IDLE
       End if
```

```
        EventType of ThisEvent is ES_JOYSTICK_UD
            START THE INTERMITTENT_STOP_TIMER
            Set CurrentState to ONE_TARGET_REACHED

       EventType of ThisEvent is ES_ALL_SPOTS_REACHED
            Set CurrentState to ALL_TARGETS_REACHED

       EventType of ThisEvent is ES_GAME_OVER
            Set CurrentState to  GAME_TIMEOUT

End MOVING block

CurrentState is ONE_TARGET_REACHED
    Based on the event received, choose one of the following blocks of code:
        EventType of ThisEvent is ES_TIMEOUT
            if EventParam of ThisEvent is IDLE_TIMER
                Set CurrentState to Spotlight_IDLE
            End If
            if EventParam of ThisEvent is  INTERMITTENT_STOP_TIMER
                Set CurrentState to Moving
            End If

End ONE_TARGET_REACHED block

CurrentState is ALL_TARGET_REACHED
    Based on the event received, choose one of the following blocks of code:
        EventType of ThisEvent is ES_REPLAY
         Post ES_NEW_MESSAGE with message parameter to LED Display Service
            Set CurrentState to BRIGHTNESS_LEVEL_CONTROL
```

EventType of ThisEvent is ES_TIMEOUT

if EventParam of ThisEvent is IDLE_TIMER

Set CurrentState to Spotlight_IDLE

End If

End GAME_TIMEOUT block

CurrentState is ALL_TARGET_REACHED

Based on the event received, choose one of the following blocks of code:

EventType of ThisEvent is ES_TIMEOUT

if EventParam of ThisEvent is IDLE_TIMER

Set CurrentState to Spotlight_IDLE

End If

EventType of ThisEvent is ES_REPLAY

Post ES_NEW_MESSAGE with message parameter to LED Display Service

Set CurrentState to BRIGHTNESS_LEVEL_CONTROL

End GAME_TIMEOUT block

Return ES_NO_EVENT

End of RunSpotlightControl

PostSpotlightControl & QuerySpotlightcontrol are not changed from the tempelate

Bool CheckBrightButton(void)

Takes the current LED index, returns nothing

Declare a variable to store CurrentButtonState

Declare a variable to store the event to post

Read current bright button state and store it

If button state has changed

If current button state is pressed

Return this event

Reset IDLE_TImer

Post ES_BRIGHT_BUTTON event to Spotlight Control module and EDMoveSenseModule

End if

End if

Set last bright button state as current state

return ReturnValue

End

Bool CheckDimButton(void)

Takes the current LED index, returns nothing

Declare a variable to store CurrentButtonState

Declare a variable to store the event to post

Read current dim button state and store it

If button state has changed

If current button state is pressed

Return this event

Reset IDLE_TImer

Post ES_DIMT_BUTTON event to Spotlight Control module and EDMoveSenseModule

End if

End if

Set last dim button state as current state

return ReturnValue

End