

```

/*----- Module Code -----*/
/*****
Function
    InitLEDDisplay

Parameters
    uint8_t : the priority of this service

Returns
    bool, false if error in initialization, true otherwise

Description
    Saves away the priority, sets up the initial transition and does any
    other required initialization for this state machine

Notes

Author
    J. Edward Carryer, 10/23/11, 18:55
*****/
bool InitLEDDisplay(uint8_t Priority)
{

    // put us into the Initial PseudoState
    // post the initial transition event

    //INITIALIZE SPI
    //disable SPI
    //use PBCLK

    //master mode, disable the analog function of B14, set B14 to an output // B14 = SCK

    //disable analog function of A0, set RA0 to an output, RA0 to the SS output, master
mode slave select control on

    //disable the analog function of A1, set A1 to an output, RA1 to the SD01

    //16 bit transfer
    //CKE to 1
    //CKP to 1
    //Enhanced buffer on
    //bit rate to 10kHz
    //enable SPI
    //clear initial flag

}

/*****
Function
    RunLEDDisplay

Parameters
    ES_Event_t : the event to process

Returns
    ES_Event_t, ES_NO_EVENT if no error ES_ERROR otherwise

Description
    add your description here

Notes

```

uses nested switch/case to implement the machine.

Author

J. Edward Carryer, 01/15/12, 15:23

*****/

ES_Event_t RunLEDDisplay(ES_Event_t ThisEvent)

{

 // assume no errors

 switch (CurrentState)

 {

 // If current state is initial Psedudo State

 // only respond to ES_Init

 //if display update steps not complete

 //Post Event

 //display is setup, put main service in welcome mode with welcome event

 // now put the machine into the actual waiting state

 //waiting state

 //If frame has been updated

 //post update row event

 //Decide what the next state will be

 //updating display state

 //update row event

 //if not done updating displays

 //re-post same event

 //if done updating displays

 //post completion event

 //update complete event

 //Decide what the next state will be

}