

```

/*****
Module
    Button.c

Revision
    1.0.1

Description
    This is a template file for implementing flat state machines under the
    Gen2 Events and Services Framework.

Notes

History
When          Who          What/Why
-----
11/04/12 23:55 ddg          create Button.c from TemplateFSM.c
01/15/12 11:12 jec          revisions for Gen2 framework
11/07/11 11:26 jec          made the queue static
10/30/11 17:59 jec          fixed references to CurrentEvent in RunTemplateSM()
10/23/11 18:20 jec          began conversion from SMTemplate.c (02/20/07 rev)
*****/
/*----- 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_ServiceHeaders.h"
#include <stdio.h>
#include <stdlib.h>

/*----- Module Defines -----*/

#define QUEUE_SIZE 10

/*----- Module Functions -----*/
/* prototypes for private functions for this machine.They should be functions
   relevant to the behavior of this state machine
*/
//ButtonState_t CheckButtonEvents ( void );
/*----- Module Variables -----*/
// everybody needs a state variable, you may need others as well.
// type of state variable should match htat of enum in header file
static unsigned char LastButtonState;
static unsigned char CurrentButtonState;
static ButtonState_t CurrentState;

static uint8_t MyPriority;

// with the introduction of Gen2, we need a module level Priority var as well

/*----- Module Code -----*/

```

```
/******
```

Function

InitializeButton

Parameters

uint8_t : the priority of this service

Returns

boolean, False if error in initialization, True otherwise

Description

Takes a priority number, returns True.

Notes

Author

Dustin D. Gerrard 11/5/12

```
*****/
```

```
boolean InitializeButton ( uint8_t Priority )
```

```
{
```

```
    // Initialize the MyPriority variable with the passed in parameter.
```

```
    MyPriority = Priority;
```

```
    // Initialize the port line to monitor the button.
```

```
    // Sample the current button state and use it to initialize LastButtonState.
```

```
    CurrentButtonState = 0;
```

```
    LastButtonState = CurrentButtonState;
```

```
    // Set CurrentState to be DEBOUNCING.
```

```
    CurrentState = DEBOUNCING;
```

```
    // Start debounce timer (timer posts to ButtonDebounceSM).
```

```
    // End of InitializeButton (return True).
```

```
    return True;
```

```
}
```

```
/******
```

Function

PostButtonFSM

Parameters

EF_Event ThisEvent , the event to post to the queue

Returns

boolean False if the Enqueue operation failed, True otherwise

Description

Posts an event to this state machine's queue

Notes

Author

Dustin D. Gerrard 11/5/12

*****/

```
boolean PostButtonSM( ES_Event ThisEvent )
```

```
{
    return ES_PostToService( MyPriority, ThisEvent);
}
```

Function

ButtonDebounceSM

Parameters

ES_Event : the event to process

Returns

ES_Event, ES_NO_EVENT if no error ES_ERROR otherwise

Description

ButtonDebounceSM (implements a 2-state state machine for debouncing timing).

Notes

Author

Dustin D. Gerrard 11/5/12

*****/

```
ES_Event ButtonDebounceSM( ES_Event ThisEvent )
```

```
{
    // Declare variables.
    ES_Event ReturnEvent;
    ReturnEvent.EventType = ES_NO_EVENT;

    // If EventType is ES_TIMEOUT & parameter is debounce timer number.

    /* ----- Button 1 -----*/
    if (ThisEvent.EventType == ES_TIMEOUT && ThisEvent.EventParam == 1)
    {
        // Set CurrentState to READY2SAMPLE.
        CurrentState = READY2SAMPLE;

    } else {
        if (ThisEvent.EventType == START_DEBOUNCE && ThisEvent.EventParam == 1)
        {
            // Start debounce timer.
            ES_Timer_SetTimer(1,50);
            ES_Timer_StartTimer(1);

            // Set CurrentState to DEBOUNCING
            CurrentState = DEBOUNCING;
        }
    }

    return ReturnEvent;
}
```

```
void ResetButtons(void)
{
    CurrentState = READY2SAMPLE;
}
```

/******

Function
 QueryButtonSM

Parameters
 None

Returns
 ButtonState_t The current state of the Template state machine

Description
 returns the current state of the Button state machine

Notes

Author

*****/

/******

private functions

*****/

ButtonState_t QueryButtonSM (void)

```
{
    return(CurrentState);
}
```