```
#include "ES_Configure.h"
#include "ES_Framework.h"
#include "ES_DeferRecall.h"
#include "ES_ShortTimer.h"

#include "inc/hw_memmap.h"
#include "inc/hw_types.h"
#include "inc/hw_gpio.h"
#include "inc/hw_sysctl.h"
#include "driverlib/sysctl.h"
#include "driverlib/pin_map.h"
#include "driverlib/gpio.h"

#include "FarmerMasterSM.h"
#include "Reverse_ButtonDebounce.h"
#include "Constants.h"

static uint8_t MyPriority;
static REV_DBState_t CurrentState;

bool InitREV_ButtonDebounce(uint8_t Priority) {
	// Set service priority
	MyPriority=Priority;
	// Initialize the current state machine state
	CurrentState = REV_Debouncing;
	// Initialize the debouncing timer
	ES_Timer_InitTimer(REV_DEBOUNCE_TIMER, DEBOUNCE_TIME);
	// End Initialization
	ES_Event ThisEvent;
	ThisEvent.EventType = ES_INIT;
	return (ES_PostToService(MyPriority, ThisEvent));
}

bool PostREV_ButtonDebounce(ES_Event ThisEvent) {
	return ES_PostToService(MyPriority, ThisEvent);
}

ES_Event RunREV_ButtonDebounce(ES_Event ThisEvent) {
	ES_Event ReturnEvent;
	ReturnEvent.EventType = ES_NO_EVENT;

	switch (CurrentState)
	{
		case (REV_Debouncing):
		{
			if ((ThisEvent.EventType == ES_TIMEOUT) && (ThisEvent.EventParam
== REV_DEBOUNCE_TIMER))
			{
				CurrentState = REV_Ready2Sample;
			}
			break;
		}

		case (REV_Ready2Sample):
		{
			if (ThisEvent.EventType == ES_BUTTON_UP)
			{
				ES_Timer_InitTimer(REV_DEBOUNCE_TIMER, DEBOUNCE_TIME);
				CurrentState = REV_Debouncing;
				ES_Event Event2Post;
				Event2Post.EventType = ES_REV_BUTTON_UP;
				PostFarmerMasterSM(Event2Post);
				printf("REV Button Up\r\n");
			}
```

```
                else if (ThisEvent.EventType == ES_BUTTON_DOWN)
                {
                      ES_Timer_InitTimer(REV_DEBOUNCE_TIMER, DEBOUNCE_TIME);
                      CurrentState = REV_Debouncing;
                      ES_Event Event2Post;
                      Event2Post.EventType = ES_REV_BUTTON_DOWN;
                      PostFarmerMasterSM(Event2Post);
                      printf("REV Button Down\r\n");
                }
                break;
          }
    }
    return ReturnEvent;
}
```