```
/****************************************************************************
 Module
     ES_Configure.h
 Description
     This file contains macro definitions that are edited by the user to
     adapt the Events and Services framework to a particular application.
 History
 When           Who     What/Why
 -------------- ---     --------
 10/21/13 20:54 jec      lots of added entries to bring the number of timers
                         and services up to 16 each
 08/06/13 14:10 jec      removed PostKeyFunc stuff since we are moving that
                         functionality out of the framework and putting it
                         explicitly into the event checking functions
 01/15/12 10:03 jec      started coding
****************************************************************************/

#ifndef CONFIGURE_H
#define CONFIGURE_H

/****************************************************************************/
// The maximum number of services sets an upper bound on the number of
// services that the framework will handle. Reasonable values are 8 and 16
// corresponding to an 8-bit(uint8_t) and 16-bit(uint16_t) Ready variable size
#define MAX_NUM_SERVICES 16

/****************************************************************************/
// This macro determines that nuber of services that are *actually* used in
// a particular application. It will vary in value from 1 to MAX_NUM_SERVICES
#define NUM_SERVICES 5

/****************************************************************************/
// These are the definitions for Service 0, the lowest priority service.
// Every Events and Services application must have a Service 0. Further
// services are added in numeric sequence (1,2,3,...) with increasing
// priorities
// the header file with the public function prototypes
#define SERV_0_HEADER "MapKeys.h"
// the name of the Init function
#define SERV_0_INIT InitMapKeys
// the name of the run function
#define SERV_0_RUN RunMapKeys
// How big should this services Queue be?
#define SERV_0_QUEUE_SIZE 3

/****************************************************************************/
// The following sections are used to define the parameters for each of the
// services. You only need to fill out as many as the number of services
// defined by NUM_SERVICES
/****************************************************************************/
// These are the definitions for Service 1
#if NUM_SERVICES > 1
// the header file with the public function prototypes
#define SERV_1_HEADER "MasterVehicle.h"
// the name of the Init function
#define SERV_1_INIT InitMasterVehicleSM
// the name of the run function
#define SERV_1_RUN RunMasterVehicleSM
// How big should this services Queue be?
#define SERV_1_QUEUE_SIZE 3
#endif
```

```
/****************************************************************************/
// These are the definitions for Service 2
#if NUM_SERVICES > 2
// the header file with the public function prototypes
#define SERV_2_HEADER "Bumper_Service.h"
// the name of the Init function
#define SERV_2_INIT  InitializeButtonDebounce
// the name of the run function
#define SERV_2_RUN RunButtonDebounceSM
// How big should this services Queue be?
#define SERV_2_QUEUE_SIZE 3
#endif

/****************************************************************************/
// These are the definitions for Service 3
#if NUM_SERVICES > 3
// the header file with the public function prototypes
#define SERV_3_HEADER "LOCMaster.h"
// the name of the Init function
#define SERV_3_INIT InitLOCMasterSM
// the name of the run function
#define SERV_3_RUN RunLOCMasterSM
// How big should this services Queue be?
#define SERV_3_QUEUE_SIZE 3
#endif

/****************************************************************************/
// These are the definitions for Service 4
#if NUM_SERVICES > 4
// the header file with the public function prototypes
#define SERV_4_HEADER "Ultrasonic.h"
// the name of the Init function
#define SERV_4_INIT InitUltrasonic
// the name of the run function
#define SERV_4_RUN RunUltrasonic
// How big should this services Queue be?
#define SERV_4_QUEUE_SIZE 3
#endif

/****************************************************************************/
// These are the definitions for Service 5
#if NUM_SERVICES > 5
// the header file with the public function prototypes
#define SERV_5_HEADER "TestHarnessService5.h"
// the name of the Init function
#define SERV_5_INIT InitTestHarnessService5
// the name of the run function
#define SERV_5_RUN RunTestHarnessService5
// How big should this services Queue be?
#define SERV_5_QUEUE_SIZE 3
#endif

/****************************************************************************/
// These are the definitions for Service 6
#if NUM_SERVICES > 6
// the header file with the public function prototypes
#define SERV_6_HEADER "TestHarnessService6.h"
// the name of the Init function
#define SERV_6_INIT InitTestHarnessService6
// the name of the run function
```

```
#define SERV_6_RUN RunTestHarnessService6
// How big should this services Queue be?
#define SERV_6_QUEUE_SIZE 3
#endif

/****************************************************************************/
// These are the definitions for Service 7
#if NUM_SERVICES > 7
// the header file with the public function prototypes
#define SERV_7_HEADER "TestHarnessService7.h"
// the name of the Init function
#define SERV_7_INIT InitTestHarnessService7
// the name of the run function
#define SERV_7_RUN RunTestHarnessService7
// How big should this services Queue be?
#define SERV_7_QUEUE_SIZE 3
#endif

/****************************************************************************/
// These are the definitions for Service 8
#if NUM_SERVICES > 8
// the header file with the public function prototypes
#define SERV_8_HEADER "TestHarnessService8.h"
// the name of the Init function
#define SERV_8_INIT InitTestHarnessService8
// the name of the run function
#define SERV_8_RUN RunTestHarnessService8
// How big should this services Queue be?
#define SERV_8_QUEUE_SIZE 3
#endif

/****************************************************************************/
// These are the definitions for Service 9
#if NUM_SERVICES > 9
// the header file with the public function prototypes
#define SERV_9_HEADER "TestHarnessService9.h"
// the name of the Init function
#define SERV_9_INIT InitTestHarnessService9
// the name of the run function
#define SERV_9_RUN RunTestHarnessService9
// How big should this services Queue be?
#define SERV_9_QUEUE_SIZE 3
#endif

/****************************************************************************/
// These are the definitions for Service 10
#if NUM_SERVICES > 10
// the header file with the public function prototypes
#define SERV_10_HEADER "TestHarnessService10.h"
// the name of the Init function
#define SERV_10_INIT InitTestHarnessService10
// the name of the run function
#define SERV_10_RUN RunTestHarnessService10
// How big should this services Queue be?
#define SERV_10_QUEUE_SIZE 3
#endif

/****************************************************************************/
// These are the definitions for Service 11
#if NUM_SERVICES > 11
// the header file with the public function prototypes
```

```
#define SERV_11_HEADER "TestHarnessService11.h"
// the name of the Init function
#define SERV_11_INIT InitTestHarnessService11
// the name of the run function
#define SERV_11_RUN RunTestHarnessService11
// How big should this services Queue be?
#define SERV_11_QUEUE_SIZE 3
#endif

/****************************************************************************/
// These are the definitions for Service 12
#if NUM_SERVICES > 12
// the header file with the public function prototypes
#define SERV_12_HEADER "TestHarnessService12.h"
// the name of the Init function
#define SERV_12_INIT InitTestHarnessService12
// the name of the run function
#define SERV_12_RUN RunTestHarnessService12
// How big should this services Queue be?
#define SERV_12_QUEUE_SIZE 3
#endif

/****************************************************************************/
// These are the definitions for Service 13
#if NUM_SERVICES > 13
// the header file with the public function prototypes
#define SERV_13_HEADER "TestHarnessService13.h"
// the name of the Init function
#define SERV_13_INIT InitTestHarnessService13
// the name of the run function
#define SERV_13_RUN RunTestHarnessService13
// How big should this services Queue be?
#define SERV_13_QUEUE_SIZE 3
#endif

/****************************************************************************/
// These are the definitions for Service 14
#if NUM_SERVICES > 14
// the header file with the public function prototypes
#define SERV_14_HEADER "TestHarnessService14.h"
// the name of the Init function
#define SERV_14_INIT InitTestHarnessService14
// the name of the run function
#define SERV_14_RUN RunTestHarnessService14
// How big should this services Queue be?
#define SERV_14_QUEUE_SIZE 3
#endif

/****************************************************************************/
// These are the definitions for Service 15
#if NUM_SERVICES > 15
// the header file with the public function prototypes
#define SERV_15_HEADER "TestHarnessService15.h"
// the name of the Init function
#define SERV_15_INIT InitTestHarnessService15
// the name of the run function
#define SERV_15_RUN RunTestHarnessService15
// How big should this services Queue be?
#define SERV_15_QUEUE_SIZE 3
#endif
```

```
/*****************************************************************************/
// Name/define the events of interest
// Universal events occupy the lowest entries, followed by user-defined events
typedef enum {  ES_NO_EVENT = 0,
                ES_ERROR,  /* used to indicate an error from the service */
                ES_INIT,   /* used to transition from initial pseudo-state */
                ES_NEW_KEY, /* signals a new key received from terminal */
                ES_TIMEOUT, /* signals that the timer has expired */
                ES_ENTRY,
                ES_ENTRY_HISTORY,
                ES_EXIT,
                /* User-defined events start here */
                CONSTRUCTION_START,
                CONSTRUCTION_END,
                STAGE_ACTIVE,
                ARRIVED_AT_STAGING,
                RESTART_VERIFY_FREQ,
                SHOOT_ACTIVE,
                SHOOT_ACTIVE_4,
                NO_BALL,
                ES_ACK,
                ES_NACK,
                ES_INACTIVE,
                SCORE_CHANGED,
                LOADED_COMPLETE,
                FINISHED_STAGING,
                X_REACHED,
                Y_REACHED,
                DBButtonDown,
                DBButtonUp,
                BUMPER_PRESSED,
                ES_LOCK,
                ES_UNLOCK,
                ES_SHORT_TIMEOUT,
                ES_EOT,//this one is end of 5 bytes
                ES_GAME_STATUS_UPDATE,
                Input_Freq,
                SEND_FIRST_REPORT,
                SEND_SECOND_REPORT,
                GO_QUERY_REPORT_RESPONSE,
                /* These events are for debuggging */
                Set_SentOne_Flag_1,
                Set_SentOne_Flag_0,
                Set_SentTwo_Flag_1,
                Set_SentTwo_Flag_0,
                LOCATION_REACHED,
                SEND_FIRST_REPORT_WITH_OUR_FAKE_FREQ,
                SEND_SECOND_REPORT_WITH_OUR_FAKE_FREQ,
                Set_first_ack_1,
                Set_first_ack_0,
                Set_second_ack_1,
                Set_second_ack_0,
                Set_Data3_Ready,
                Set_Data3_Notready,
                Set_Data4_ACK,
                Set_Data4_NACK
   } ES_EventTyp_t ;

/*****************************************************************************/
// These are the definitions for the Distribution lists. Each definition
```

```
// should be a comma separated list of post functions to indicate which
// services are on that distribution list.
#define NUM_DIST_LISTS 4
#if NUM_DIST_LISTS > 0
#define DIST_LIST0 PostMasterVehicleSM, PostMasterVehicleSM
#endif
#if NUM_DIST_LISTS > 1
#define DIST_LIST1 PostMapKeys, PostMapKeys
#endif
#if NUM_DIST_LISTS > 2
#define DIST_LIST2 PostButtonDebounceService, PostButtonDebounceService
#endif
#if NUM_DIST_LISTS > 3
#define DIST_LIST3 PostUltrasonic, PostUltrasonic
#endif
#if NUM_DIST_LISTS > 4
#define DIST_LIST4 PostTemplateFSM
#endif
#if NUM_DIST_LISTS > 5
#define DIST_LIST5 PostTemplateFSM
#endif
#if NUM_DIST_LISTS > 6
#define DIST_LIST6 PostTemplateFSM
#endif
#if NUM_DIST_LISTS > 7
#define DIST_LIST7 PostTemplateFSM
#endif

/****************************************************************************/
// This are the name of the Event checking funcion header file.
#define EVENT_CHECK_HEADER "EventCheckers.h"

/****************************************************************************/
// This is the list of event checking functions
#define EVENT_CHECK_LIST Check4Keystroke, CheckButtonEvents, Location_Checker

/****************************************************************************/
// These are the definitions for the post functions to be executed when the
// corresponding timer expires. All 16 must be defined. If you are not using
// a timer, then you should use TIMER_UNUSED
// Unlike services, any combination of timers may be used and there is no
// priority in servicing them
#define TIMER_UNUSED ((pPostFunc)0)
#define TIMER0_RESP_FUNC PostMasterVehicleSM
#define TIMER1_RESP_FUNC PostMasterVehicleSM
#define TIMER2_RESP_FUNC PostMasterVehicleSM
#define TIMER3_RESP_FUNC PostMasterVehicleSM
#define TIMER4_RESP_FUNC PostMasterVehicleSM
#define TIMER5_RESP_FUNC PostMasterVehicleSM
#define TIMER6_RESP_FUNC PostMasterVehicleSM
#define TIMER7_RESP_FUNC PostMasterVehicleSM
#define TIMER8_RESP_FUNC PostMasterVehicleSM
#define TIMER9_RESP_FUNC PostLOCMasterSM
#define TIMER10_RESP_FUNC PostLOCMasterSM
#define TIMER11_RESP_FUNC PostMasterVehicleSM
#define TIMER12_RESP_FUNC PostButtonDebounceService
#define TIMER13_RESP_FUNC PostMapKeys
#define TIMER14_RESP_FUNC PostMasterVehicleSM
#define TIMER15_RESP_FUNC PostUltrasonic
```

```
/*****************************************************************************/
// Give the timer numbers symbolc names to make it easier to move them
// to different timers if the need arises. Keep these definitions close to the
// definitions for the response functions to make it easier to check that
// the timer number matches where the timer event will be routed
// These symbolic names should be changed to be relevant to your application

#define SERVICE0_TIMER 15
#define STAGE_TIMER 14
#define DISPLAY_TIMER 13
#define BUTTON_TIMER 12
#define FREQ_MEASURE_TIMER 11
#define GET_STATUS_TIMER 10
#define REPORT_TIMER 9
#define SHOOT_20S_TIMER 8
#define SHOT_DELAY_TIMER 7
#define SHOOTER_RESET_TIMER 6
#define READY_TO_SHOOT_TIMER 5
#define LATCH_DISPENSE_TIMER 4
#define START_SHOOTER_TIMER 3
#define SUPPLY_LED_TIMER 2
#define SUPPLY_TIMER 1
#define SUPPLY_RAMMING_TIMER 0
#endif /* CONFIGURE_H */
```