

```

/*****
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.
Notes

History
When          Who          What/Why
-----
01/15/12 10:03 jec          started coding
02/05/13 11:45 YtZ          Started Modification
*****/

#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
// HOWEVER: at this time only a value of 8 is supported.
#define MAX_NUM_SERVICES 8

/*****/
// 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 6

/*****/
// 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 fuction prototypes
#define SERV_0_HEADER "CommandSM.h"
// the name of the Init function
#define SERV_0_INIT InitCommandSM
// the name of the run function
#define SERV_0_RUN RunCommandSM
// 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 fuction prototypes
#define SERV_1_HEADER "PltfmFSM.h"
// the name of the Init function
#define SERV_1_INIT InitPltfmFSM

```

```

// the name of the run function
#define SERV_1_RUN RunPltfrmFSM
// 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 fuction prototypes
#define SERV_2_HEADER "SPIService.h"
// the name of the Init function
#define SERV_2_INIT InitSPIService
// the name of the run function
#define SERV_2_RUN RunSPIService
// 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 fuction prototypes
#define SERV_3_HEADER "ShootService.h"
// the name of the Init function
#define SERV_3_INIT InitShootService
// the name of the run function
#define SERV_3_RUN RunShootService
// 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 fuction prototypes
#define SERV_4_HEADER "SignalService.h"
// the name of the Init function
#define SERV_4_INIT InitSignalService
// the name of the run function
#define SERV_4_RUN RunSignalService
// 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 fuction prototypes
#define SERV_5_HEADER "MasterTimerService.h"
// the name of the Init function
#define SERV_5_INIT InitMasterTimerService
// the name of the run function
#define SERV_5_RUN RunMasterTimerService

```

```

// 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 fuction prototypes
#define SERV_6_HEADER "TestService.h"
// the name of the Init function
#define SERV_6_INIT TestServiceInit
// the name of the run function
#define SERV_6_RUN TestServiceRun
// 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 fuction prototypes
#define SERV_7_HEADER "TestService.h"
// the name of the Init function
#define SERV_7_INIT TestServiceInit
// the name of the run function
#define SERV_7_RUN TestServiceRun
// How big should this services Queue be?
#define SERV_7_QUEUE_SIZE 3
#endif

/*****/
// the name of the posting function that you want executed when a new
// keystroke is detected.
// The default initialization distributes keystrokes to all state machines
#define POST_KEY_FUNC ES_PostAll

/*****/
// 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 */
               /* User-defined events start here */
               ES_INTERRUPT,
               ES_NEW_COMMAND,
               ES_INITIALIZE,
               ES_STOP,
               ES_RCW,
               ES_RCCW,
               ES_RCW_90,
               ES_RCW_45,
               ES_RCCW_90,

```

```
ES_RCCW_45,  
ES_FWD_FS,  
ES_FWD_HS,  
ES_BWD_FS,  
ES_BWD_HS,  
ES_BEACON,  
ES_TAPE,  
ES_BEACON_DETECTED,  
ES_TAPE_DETECTED,  
ES_Command,  
ES_STATUS,  
ES_Bytereceived,  
ES_GAMEPLAY,  
ES_TGT_FOUND_XX,  
ES_TGT_FOUND_YY,  
ES_TGT_FOUND_RR,  
ES_BOT_DETECTED,  
ES_SHOOT,  
ES_RELOAD,  
ES_MASTER_TIMEOUT,  
ES_SHOOT_CALIBRATE,  
ES_SHOOT_DONE  
  
} 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 0  
#if NUM_DIST_LISTS > 0  
#define DIST_LIST0 //PostMotorFSM  
#endif  
#if NUM_DIST_LISTS > 1  
#define DIST_LIST1 //PostEncoderService  
#endif  
#if NUM_DIST_LISTS > 2  
#define DIST_LIST2 PostTemplateFSM  
#endif  
#if NUM_DIST_LISTS > 3  
#define DIST_LIST3 PostTemplateFSM  
#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 function header file.
#define EVENT_CHECK_HEADER "EventCheckers.h"

/*****/
// This is the list of event checking functions
#define EVENT_CHECK_LIST Check_for_Tape

/*****/
// These are the definitions for the post functions to be executed when the
// correspondng timer expires. All 8 must be defined. If you are not using
// a timers, then you can use TIMER_UNUSED
#define TIMER_UNUSED ((pPostFunc)0)
#define TIMER0_RESP_FUNC PostCommandSM
#define TIMER1_RESP_FUNC PostPltfmFSM
#define TIMER2_RESP_FUNC PostShootService
#define TIMER3_RESP_FUNC PostSignalService
#define TIMER4_RESP_FUNC PostMasterTimerService
#define TIMER5_RESP_FUNC TIMER_UNUSED
#define TIMER6_RESP_FUNC TIMER_UNUSED
#define TIMER7_RESP_FUNC TIMER_UNUSED

/*****/
// Give the timer numbers symbolc names to make it easier to move them
// to different timers if the need arises. Keep these definitons 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

#define SPICommandTimer 1
#define PltfmTimer 1
#define CommandTimer 0
#define ShootTimer 2
#define SignalTimer 3
#define MasterTimer 4

#endif /* CONFIGURE_H */

```