

```

/*****

```

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
-----	---	-----
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
11/18/13	whg	started converting to our project

```

*****/

```

```

#ifndef CONFIGURE_H
#define CONFIGURE_H
#include <mc9s12c32.h>
#include <stdio.h>
#include "bitdefs.h"

```

```

#define True      1
#define False     0

```

```

/*****/

```

```

// 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 number of services that are *actually* used in
// a particular application. It will vary in value from 1 to MAX_NUM_SERVICES
#define NUM_SERVICES 8

```

```

/*****/

```

```

// 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 "GameFSM.h"

```

```

// the name of the Init function

```

```

#define SERV_0_INIT InitGameFSM

```

```

// the name of the run function

```

```

#define SERV_0_RUN RunGameFSM

```

```

// How big should this services Queue be?

```

```

#define SERV_0_QUEUE_SIZE 3

```

```

/*****/
// These are the definitions for Service 1
#if NUM_SERVICES > 1
// the header file with the public function prototypes
#define SERV_1_HEADER "TargetFSM.h"
// the name of the Init function
#define SERV_1_INIT InitTargetFSM
// the name of the run function
#define SERV_1_RUN RunTargetFSM
// 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 "Debounce.h"
// the name of the Init function
#define SERV_2_INIT InitDebounce
// the name of the run function
#define SERV_2_RUN RunDebounce
// 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 "VibrationControl.h"
// the name of the Init function
#define SERV_3_INIT InitVibrationControl
// the name of the run function
#define SERV_3_RUN RunVibrationControl
// 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 "TargetControl.h"
// the name of the Init function
#define SERV_4_INIT InitTargetControl
// the name of the run function
#define SERV_4_RUN RunTargetControl
// 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 "BalanceFSM.h"
// the name of the Init function
#define SERV_5_INIT InitBalanceFSM
// the name of the run function
#define SERV_5_RUN RunBalanceFSM
// 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 "CommFSM.h"
// the name of the Init function
#define SERV_6_INIT InitCommFSM
// the name of the run function
#define SERV_6_RUN RunCommFSM
// How big should this services Queue be?
#define SERV_6_QUEUE_SIZE 5
#endif

/*****/
// These are the definitions for Service 7
#if NUM_SERVICES > 7
// the header file with the public fuction prototypes
#define SERV_7_HEADER "ClockControl.h"
// the name of the Init function
#define SERV_7_INIT InitClockControl
// the name of the run function
#define SERV_7_RUN RunClockControl
// 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 fuction 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 fuction 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 fuction 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 fuction 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 fuction 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 fuction 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 fuction 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 fuction 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 */

               /* User-defined events start here */
               ES_ButtonDown, /* button has been pressed */
               ES_AccessCardDetected, /* access card has been detected */
               ES_AccessCardRemoved, /* access card has been removed */
               ES_SEND_REQUEST, /* new message waiting to be sent */
               ES_STROBE_RISE, /* new message transmission has started */

```

```

ES_STROBE_FALL, /* new message transmission has ended */
ES_ACK_RISE, /* new message reception has started */
ES_ACK_FALL, /* new message reception has ended */
ES_WentOffBalance, /* board has become unbalanced */
ES_WentOnBalance, /* board has become balanced */
ES_BalanceChange, /* board position has changed */
ES_ACovered, /* active target becomes covered by a hand */
ES_AUncovered, /* active target becomes uncovered */

```

```

/* Internal Events */
ES_BothRebalanced,
ES_Unbalanced,
ES_BothCovered,
ES_Begin,
ES_BReadyPush,
ES_BCovered,
ES_BUncovered,
ES_BUnbalanced,
ES_BRebalanced,
ES_START_CLOCK
} ES_EventTyp_t ;

```

```

/*****

```

```

// These are the definitions for the Distribution lists. Each definition
// should be a comma seperated list of post functions to indicate which
// services are on that distribution list.

```

```

#define NUM_DIST_LISTS 3
#if NUM_DIST_LISTS > 0
#define DIST_LIST0 PostGameFSM, PostBalanceFSM, PostTargetFSM
#endif
#if NUM_DIST_LISTS > 1
#define DIST_LIST1 PostBalanceFSM, PostTargetFSM
#endif
#if NUM_DIST_LISTS > 2
#define DIST_LIST2 PostGameFSM, PostBalanceFSM
#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 funcion header file.
#define EVENT_CHECK_HEADER "EventCheckers.h"

/*****/
// This is the list of event checking functions
#define EVENT_CHECK_LIST Check4Strobe, Check4Ack, CheckForTargetCovered, CheckForTiltChange

/*****/
// These are the definitions for the post functions to be executed when the
// correspnding 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 PostDebounce
#define TIMER1_RESP_FUNC PostTargetControl
#define TIMER2_RESP_FUNC ES_PostList00 //game timer
#define TIMER3_RESP_FUNC PostGameFSM // powerdown timeout
#define TIMER4_RESP_FUNC PostGameFSM
#define TIMER5_RESP_FUNC PostGameFSM
#define TIMER6_RESP_FUNC PostCommFSM
#define TIMER7_RESP_FUNC PostTargetControl
#define TIMER8_RESP_FUNC PostTargetControl
#define TIMER9_RESP_FUNC PostClockControl
#define TIMER10_RESP_FUNC PostTargetControl
#define TIMER11_RESP_FUNC TIMER_UNUSED
#define TIMER12_RESP_FUNC TIMER_UNUSED
#define TIMER13_RESP_FUNC TIMER_UNUSED
#define TIMER14_RESP_FUNC TIMER_UNUSED
#define TIMER15_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
// These symbolic names should be changed to be relevant to your application

#define DEBOUNCE_TIMER          0
#define DEBOUNCE_TIME          20 // 20 ms
// post this timer to:

#define TARGET_PULSE_TIMER      1
#define TARGET_PULSE_TIME      150 // 150 ms
// post this timer to:

// UNUSED IN B
#define GAME_TIMER              2
#define GAME_TIME               35000 // 30 s
// post this timer to:

```

```

// UNUSED IN B

#define POWER_DOWN_TIMER          3
#define POWER_DOWN_TIME          5000 // 5 s
// post this timer to:

#define BALANCE_TIMER             4
#define BALANCE_TIME              4000 // 4 s
// post this timer to:

#define TARGET_TIMER              5
#define TARGET_TIME                4000 // 4 s
// post this timer to:

#define COMM_TIMER                6
#define COMM_TIME                  90 //10 ms
// post this timer to:

#define WAITING_SHOW_TIMER        7
#define WAITING_SHOW_TIME          500 // 0.5 s
// post this timer to: PostTargetControl

#define END_SHOW_TIMER            8
#define END_SHOW_TIME              100 // ~200Hz
// post this timer to: PostTargetControl

#define CLOCK_TIMER               9
#define CLOCK_TIME                 (int) (GAME_TIME/8)

#define BLINK_TIMER               10
#define BLINK_TIME                 END_SHOW_TIME * 2

/*****/
// Pin assignments
// ----- PORT AD ----- //
#define POT_INPUT_PIN             BIT6HI // AD6
#define POT_INPUT_PORT_PIN        6 // relative pin in AD port
#define ANALOG_INITS              "IAII0000"

#define ACCESS_CARD_INPUT_PIN     BIT4HI // AD4
#define ACCESS_CARD_DIR           0 //Initialized with analog
#define ACCESS_CARD_PORT          PTIAD

#define BOARD_SWITCH_INPUT_PIN    BIT7HI //AD7
#define BOARD_SWITCH_DIR          0
#define BOARD_SWITCH_PORT         PTIAD

#define SHIFT_REG_DATA            BIT0HI //AD0
#define SHIFT_REG_SCLK            BIT1HI //AD1
#define SHIFT_REG_RCLK            BIT2HI //AD2
#define SHIFT_REG_EN              BIT3HI //AD3
#define SHIFT_REG_PORT            PTAD

```



```
// ----- PORT M ----- //

#define BUTTON_INPUT_PIN      BIT5HI // M5
#define BUTTON_DIR            DDRM
#define BUTTON_PORT           PTM

//Define Directory
#define D0_DIR                DDRM_DDRM4 //DataBit 0
#define D1_DIR                DDRM_DDRM3 //DataBit 1
#define D2_DIR                DDRM_DDRM2 //DataBit 2
#define ACK_DIR              DDRM_DDRM0 //Acknowledge Bit
#define STROBE_DIR           DDRM_DDRM1 //Strobe Bit

//Define Register
#define D0_BIT                PTM_PTM4 //DataBit 0
#define D1_BIT                PTM_PTM3 //DataBit 1
#define D2_BIT                PTM_PTM2 //DataBit 2
#define ACK_BIT              PTM_PTM0 //Acknowledge Bit
#define STROBE_BIT           PTM_PTM1 //Strobe Bit

// ----- PORT T ----- //
// Green Target Wire
#define TARGET_CONTROL_PIN    BIT0HI // T0 (PWM pin)
#define TARGET_CONTROL_GRP    PWMS12_GRP0
#define TARGET_CONTROL_CHNL   PWMS12_CHAN0
#define TARGET_CONTROL_DIR    DDRT
#define TARGET_CONTROL_PORT   PTT

// Red Target Wire
#define BRIGHTEN_CONTROL_PIN  BIT1HI // T1 (PWM pin)
#define BRIGHTEN_CONTROL_GRP  PWMS12_GRP0
#define BRIGHTEN_CONTROL_CHNL PWMS12_CHAN1
#define BRIGHTEN_CONTROL_DIR  DDRT
#define BRIGHTEN_CONTROL_PORT PTT

#define VIB_MOTOR_PIN         BIT2HI // T2 (PWM pin)
#define VIB_MOTOR_GRP         PWMS12_GRP1
#define VIB_MOTOR_CHNL        PWMS12_CHAN2
#define VIB_MOTOR_DIR         DDRT
#define VIB_MOTOR_PORT        PTT

#define TARGET_SEL_A_PIN      BIT3HI // T3
#define TARGET_SEL_A_DIR      DDRT
#define TARGET_SEL_A_PORT     PTT

#define TARGET_SEL_B_PIN      BIT4HI // T4
#define TARGET_SEL_B_DIR      DDRT
#define TARGET_SEL_B_PORT     PTT

// Black Target Wire
#define TARGET_SENSOR_PIN     BIT5HI // T5
#define TARGET_SENSOR_DIR     DDRT
#define TARGET_SENSOR_PORT    PTT
```

```
#define SERVO_INIT_STRING      "Sxxxxxxx" // T7
#define SERVO_PIN              7

// ----- PORT E ----- //

/*****/
// Servo positions (in us)

#define SERVO_0_DEG           550
//#define SERVO_30_DEG        0
#define SERVO_60_DEG          950
#define SERVO_90_DEG          1400
#define SERVO_120_DEG         1850
//#define SERVO_150_DEG       0
#define SERVO_180_DEG         2450

#endif /* CONFIGURE_H */
```