

```

/**
 * ES_Configure.h
 * This file contains macro definitions that are edited by the user to
 * adapt the Events and Services framework to a particular application.
 * This version is a modification of that developed by Prof. J. Edward
 * Carryer.
 */

#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 "WireTouchDB.h"
// the name of the Init function
#define SERV_0_INIT InitializeWireTouchDB
// the name of the run function
#define SERV_0_RUN RunWireTouchDB
// How big should this services Queue be?
#define SERV_0_QUEUE_SIZE 5

/*****
// 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 "GearMotor.h"
// the name of the Init function
#define SERV_1_INIT InitializeGearMotor
// the name of the run function
#define SERV_1_RUN RunGearMotor
// 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 "VibrateMotor.h"
// the name of the Init function
#define SERV_2_INIT InitializeVibrateMotor
// the name of the run function
#define SERV_2_RUN RunVibrateMotor
// 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 "Clock.h"
// the name of the Init function
#define SERV_3_INIT InitializeClock
// the name of the run function
#define SERV_3_RUN RunClock
// 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 "Mummy.h"
// the name of the Init function
#define SERV_4_INIT InitializeMummy
// the name of the run function
#define SERV_4_RUN RunMummySM
// 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_TIMEOUT, /* signals that the timer has expired */
              /* User-defined events start here */
              ES_NEW_KEY, /* signals a new key received from terminal */
              BeardRotated,
              ButtonPressed,
              ButtonReleased,
              DoorClose,
              DoorOpen,
              MiddleHallsFall,
              MotorOn,
              StartHallsFall,
              WireTouch,
              WireTouchDB } 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 1
#if NUM_DIST_LISTS > 0
#define DIST_LIST0 PostMummySM
#endif
#if NUM_DIST_LISTS > 1
#define DIST_LIST1 PostTestHarnessService1, PostTestHarnessService1
#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 is 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 Check4Keystroke, CheckButton1Events, CheckButton2Events, CheckButton3Events, CheckButton4Events, CheckButton5Ever

/*****
// 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 PostMummySM // Timer 0 used for DDM timer
#define TIMER1_RESP_FUNC PostMummySM // Timer 1 used for Joy timer
#define TIMER2_RESP_FUNC PostVibrateMotor // Timer 2 used for vibration motor timer
#define TIMER3_RESP_FUNC PostClock // Timer 3 used for ticking the DDM clock
#define TIMER4_RESP_FUNC PostGearMotor // Timer 4 used for timing the opening/closing of the lid/door
#define TIMER5_RESP_FUNC PostWireTouchDB // Timer 5 used for debouncing WireTouch event
#define TIMER6_RESP_FUNC TIMER_UNUSED
#define TIMER7_RESP_FUNC TIMER_UNUSED
#define TIMER8_RESP_FUNC TIMER_UNUSED
#define TIMER9_RESP_FUNC TIMER_UNUSED
#define TIMER10_RESP_FUNC TIMER_UNUSED
#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 symbolic 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 DDM_TIMER 0 // Timer 0 used for DDM timer
#define JOY_TIMER 1 // Timer 1 used for Joy timer
#define VIB_MOTOR_TIMER 2 // Timer 2 used for vibration motor timer
#define CLOCK_TIMER 3 // Timer 3 used for ticking the DDM clock
#define GEAR_MOTOR_TIMER 4 // Timer 4 used for timing the opening/closing of the lid/door
#define WIRE_DB_TIMER 5 // Timer 5 used for debouncing WireTouch event

/*****
// Special macros for the Mummy DDM
#define BUTTON_1_PORT HWREG(GPIO_PORTF_BASE + (GPIO_O_DATA + ALL_BITS)) // Port F
#define BUTTON_1_PIN BIT0HI // PF0

#define BUTTON_2_PORT HWREG(GPIO_PORTE_BASE + (GPIO_O_DATA + ALL_BITS)) // Port E
#define BUTTON_2_PIN BIT2HI // PE2

#define BUTTON_3_PORT HWREG(GPIO_PORTD_BASE + (GPIO_O_DATA + ALL_BITS)) // Port D
#define BUTTON_3_PIN BIT7HI // PD7

#define BUTTON_4_PORT HWREG(GPIO_PORTE_BASE + (GPIO_O_DATA + ALL_BITS)) // Port E
#define BUTTON_4_PIN BIT4HI // PE4

#define BUTTON_5_PORT HWREG(GPIO_PORTE_BASE + (GPIO_O_DATA + ALL_BITS)) // Port E
#define BUTTON_5_PIN BIT5HI // PE5

#define BUTTON_6_PORT HWREG(GPIO_PORTF_BASE + (GPIO_O_DATA + ALL_BITS)) // Port F
#define BUTTON_6_PIN BIT1HI // PF1

#define MOTOR_1_PORT HWREG(GPIO_PORTA_BASE + (GPIO_O_DATA + ALL_BITS)) // Port A
#define MOTOR_1_PIN BIT6HI // PA6

#define MOTOR_2_PORT HWREG(GPIO_PORTA_BASE + (GPIO_O_DATA + ALL_BITS)) // Port A
#define MOTOR_2_PIN BIT7HI // PA7

#define MOTOR_3_PORT HWREG(GPIO_PORTC_BASE + (GPIO_O_DATA + ALL_BITS)) // Port C
#define MOTOR_3_PIN BIT4HI // PC4

#define MOTOR_4_PORT HWREG(GPIO_PORTC_BASE + (GPIO_O_DATA + ALL_BITS)) // Port C
#define MOTOR_4_PIN BIT5HI // PC5

```

```

#define MOTOR_5_PORT HWREG(GPIO_PORTC_BASE + (GPIO_O_DATA + ALL_BITS)) // Port C
#define MOTOR_5_PIN BIT6HI // PC6

#define MOTOR_6_PORT HWREG(GPIO_PORTC_BASE + (GPIO_O_DATA + ALL_BITS)) // Port C
#define MOTOR_6_PIN BIT7HI // PC7

#define GMOTOR_IN_1_PORT HWREG(GPIO_PORTB_BASE + (GPIO_O_DATA + ALL_BITS)) // Port B
#define GMOTOR_IN_1_PIN BIT0HI // PB0

#define GMOTOR_IN_2_PORT HWREG(GPIO_PORTB_BASE + (GPIO_O_DATA + ALL_BITS)) // Port B
#define GMOTOR_IN_2_PIN BIT1HI // PB1

#define BEARD_PIN 3 // PE0 => 3

#define HALL_1A_PORT HWREG(GPIO_PORTF_BASE + (GPIO_O_DATA + ALL_BITS)) // Port F
#define HALL_1A_PIN BIT4HI // PF4

#define HALL_1B_PORT HWREG(GPIO_PORTF_BASE + (GPIO_O_DATA + ALL_BITS)) // Port F
#define HALL_1B_PIN BIT3HI // PF3

#define HALL_2A_PORT HWREG(GPIO_PORTF_BASE + (GPIO_O_DATA + ALL_BITS)) // Port F
#define HALL_2A_PIN BIT2HI // PF2

#define HALL_2B_PORT HWREG(GPIO_PORTD_BASE + (GPIO_O_DATA + ALL_BITS)) // Port D
#define HALL_2B_PIN BIT2HI // PD2

#define WIRE_PORT HWREG(GPIO_PORTD_BASE + (GPIO_O_DATA + ALL_BITS)) // Port D
#define WIRE_PIN BIT1HI // PD1

#define WIRE_GAME_LED_PORT HWREG(GPIO_PORTD_BASE + (GPIO_O_DATA + ALL_BITS)) // Port D
#define WIRE_GAME_LED_PIN BIT6HI // PD6

#define SREG_SCLK_PORT HWREG(GPIO_PORTA_BASE + (GPIO_O_DATA + ALL_BITS)) // Port A
#define SREG_SCLK_PIN BIT2HI // PA2

#define SREG_RCLK_PORT HWREG(GPIO_PORTD_BASE + (GPIO_O_DATA + ALL_BITS)) // Port D
#define SREG_RCLK_PIN BIT3HI // PD3

#define SREG_SER_PORT HWREG(GPIO_PORTB_BASE + (GPIO_O_DATA + ALL_BITS)) // Port B
#define SREG_SER_PIN BIT2HI // PB2

#define CLOCK_PIN 0 // PB6 => 0
#define CLOCK_GROUP 0 // PB6 => 0

#define FACE_PIN 3 // PB5 => 3
#define FACE_GROUP 1 // PB5 => 1

#define EYE_RED_PORT HWREG(GPIO_PORTA_BASE + (GPIO_O_DATA + ALL_BITS)) // Port A
#define EYE_RED_PIN BIT3HI // PA3

#define EYE_GREEN_PORT HWREG(GPIO_PORTA_BASE + (GPIO_O_DATA + ALL_BITS)) // Port A
#define EYE_GREEN_PIN BIT4HI // PA4

#define EYE_BLUE_PORT HWREG(GPIO_PORTA_BASE + (GPIO_O_DATA + ALL_BITS)) // Port A
#define EYE_BLUE_PIN BIT5HI // PA5

#define ARM_STATE_PORT HWREG(GPIO_PORTE_BASE + (GPIO_O_DATA + ALL_BITS)) // Port E
#define ARM_STATE_PIN BIT3HI // PE3

#define ALL_BITS (0xff << 2) // differential address to access all bits in register at once
#define TICK 1000 // tick rate for clock servo [ms]
#define LIMIT_1 3200 // one limit of servo
#define LIMIT_2 680 // other limit of servo

#endif /* CONFIGURE_H */

```