

```

/*****
Template header file for Hierarchical Sate Machines AKA StateCharts
02/08/12 adjstments for use with the Events and Services Framework Gen2
3/17/09 Fixed prototyptes to use Event_t
*****/

#ifndef SensorService_H
#define SensorService_H

// typedefs for the states
// State definitions for use with the query function
typedef enum { SensorState } TemplateState_t ;

// Public Function Prototypes

ES_Event RunSensorService( ES_Event CurrentEvent );
TemplateState_t QuerySensorServiceSM ( void );
bool InitSensorService ( uint8_t Priority );
bool PostSensorService( ES_Event ThisEvent );

unsigned char get_cal_pot(void);
unsigned char get_pushbutton(void);
unsigned char get_dipswitch(void);
unsigned char get_dipswitch_1(void);
unsigned char get_dipswitch_2(void);
unsigned char get_dipswitch_3(void);
unsigned char get_home_fwd(void);
unsigned char get_home_rvs(void);
unsigned char get_bumper_fwd(void);
unsigned char get_bumper_rvs(void);
unsigned char get_tape_left(void);
unsigned char get_tape_right(void);
unsigned char get_tape_center(void);
uint16_t get_sonic_fwd(void);
uint16_t get_sonic_back(void);
uint16_t get_sonic_side_fwd(void);
uint16_t get_sonic_side_back(void);
void UpdateLEDs(unsigned char R, unsigned char G, unsigned char B);

#include "ES_Types.h"
#include "ES_Configure.h"
#include "ES_Timers.h"
#include "ADS12.h"
#include <hidef.h>
#include <mc9s12e128.h>
#include <Bin_Const.h>
#include <termio.h>
#include <S12eVec.h>
#include <stdio.h>
#include <S12e128bits.h> //from osagi's file, is this redundant?
#include <stdlib.h>
#include "MasterHSM.h"

#define RED_BIT BIT0HI //AD output port for red status indicator LED
#define GREEN_BIT BIT1HI //AD output port for green status indicator LED
#define BLUE_BIT BIT2HI //AD output port for blue status indicator LED

#define POT_PIN 3 //AD port used for potentiometer setpoint

```

```

#define DIPSWITCH_1 BIT4HI //AD input port for dip switch 1
#define DIPSWITCH_2 BIT5HI //AD input port for dip switch 2
#define DIPSWITCH_3 BIT6HI //AD input port for dip switch 3

#define PUSHBUTTON BIT7HI //AD input port for pushbutton

#define SENSOR_TIMER 9
#define SENSOR_INTERVAL_MS 250

//#define KNIGHT_TIMER 8
//#define _MS_ *187.5
#define BEACON_CHECK_PERIOD 19//assumes 128 divisor for 24 MHz clock, corresponds to a checking
frequency slightly slower than 10 kHz
//#define REFRESH_PERIOD (100_MS_)

#define MAX_BEACON_CHECK_INDEX 2500 //corresponds to a lost IR pulse timeout period of about
0.250 seconds

//Note: Our output pulse to the ultrasonic sensors is inverted through a MOSFET
#define ULTRASONIC_HIGH_PERIOD 2500
#define ULTRASONIC_LOW_PERIOD 4 // Send a pulse width of ~ 100 us

#define WALL_OFFSET_NOMINAL 70 //nominal target distance for our bot away from the wall,
units: mm

//#define SOCCER_OFFSET 200 //where to stop when we want to soccer
#define SLOW_DOWN_FWD_OFFSET 350
#define SLOW_DOWN_RVS_OFFSET 275
#define HOME_FWD_OFFSET 100
#define HOME_RVS_OFFSET 70
// where to stop when the round is over

//#define GOAL_DEBUG

//#define PRINT_SENSOR_CALLS //this will spam the terminal window every time the routine runs (which is
frequent if we are polling switches)
//#define PRINT_SENSOR_READS

//#define MANUAL_COMMANDS

#endif /*SensorService_H */

```