```
/*******************************************************************************
  MPLAB Harmony Application Header File

  Company:
    Microchip Technology Inc.

  File Name:
    app.h

  Summary:
    This header file provides prototypes and definitions for the application.

  Description:
    This header file provides function prototypes and data type definitions for
    the application.  Some of these are required by the system (such as the
    "APP_Initialize" and "APP_Tasks" prototypes) and some of them are only used
    internally by the application (such as the "APP_STATES" definition).  Both
    are defined here for convenience.
*******************************************************************************/

//DOM-IGNORE-BEGIN
/*******************************************************************************
Copyright (c) 2013-2014 released Microchip Technology Inc.  All rights reserved.

Microchip licenses to you the right to use, modify, copy and distribute
Software only when embedded on a Microchip microcontroller or digital signal
controller that is integrated into your product or third party product
(pursuant to the sublicense terms in the accompanying license agreement).

You should refer to the license agreement accompanying this Software for
additional information regarding your rights and obligations.

SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND,
EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF
MERCHANTABILITY, TITLE, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE.
IN NO EVENT SHALL MICROCHIP OR ITS LICENSORS BE LIABLE OR OBLIGATED UNDER
CONTRACT, NEGLIGENCE, STRICT LIABILITY, CONTRIBUTION, BREACH OF WARRANTY, OR
OTHER LEGAL EQUITABLE THEORY ANY DIRECT OR INDIRECT DAMAGES OR EXPENSES
INCLUDING BUT NOT LIMITED TO ANY INCIDENTAL, SPECIAL, INDIRECT, PUNITIVE OR
CONSEQUENTIAL DAMAGES, LOST PROFITS OR LOST DATA, COST OF PROCUREMENT OF
SUBSTITUTE GOODS, TECHNOLOGY, SERVICES, OR ANY CLAIMS BY THIRD PARTIES
(INCLUDING BUT NOT LIMITED TO ANY DEFENSE THEREOF), OR OTHER SIMILAR COSTS.
 *******************************************************************************/
//DOM-IGNORE-END

#ifndef _APP_H
#define _APP_H

// *****************************************************************************
// *****************************************************************************
// Section: Included Files
// *****************************************************************************
// *****************************************************************************

#include <stdint.h>
#include <stdbool.h>
#include <stddef.h>
#include <stdlib.h>
#include "system_config.h"
#include "system_definitions.h"
#include "bno055.h"

// DOM-IGNORE-BEGIN
#ifdef __cplusplus  // Provide C++ Compatibility

extern "C" {

#endif
// DOM-IGNORE-END

#define TIME_OUT 80000000U
#define TIME_POWER_OFF   500
```

```
73
74 // *****************************************************************************
75 // *****************************************************************************
76 // Section: Type Definitions
77 // *****************************************************************************
78 // *****************************************************************************
79 typedef struct {
80     s32 comres;
81     bool flagMeasReady;
82     uint8_t flagImportantMeas;
83     struct bno055_gravity_double_t gravity;
84     struct bno055_linear_accel_double_t linear_accel;
85     struct bno055_euler_double_t euler;
86     struct bno055_gyro_double_t gyro;
87     struct bno055_mag_double_t mag;
88     struct bno055_quaternion_t quaternion;
89     unsigned long time;
90     unsigned long l_time;
91     uint16_t d_time;
92     float pressure;
93 }s_bno055_data;
94 // *****************************************************************************
95 /* Application states
96
97   Summary:
98     Application states enumeration
99
100   Description:
101     This enumeration defines the valid application states.  These states
102     determine the behavior of the application at various times.
103 */
104
105 typedef enum
106 {
107         /* Application's state machine's initial state. */
108         APP_STATE_INIT=0,
109     APP_STATE_LOGGING,
110     APP_STATE_FLAG_MEAS,
111         APP_STATE_SHUTDOWN
112         /* TODO: Define states used by the application state machine. */
113
114 } APP_STATES;
115
116
117 // *****************************************************************************
118 /* Application Data
119
120   Summary:
121     Holds application data
122
123   Description:
124     This structure holds the application's data.
125
126   Remarks:
127     Application strings and buffers are be defined outside this structure.
128  */
129
130 typedef struct
131 {
132     /* The application's current state */
133     APP_STATES state;
134 } APP_DATA;
135
136 typedef struct
137 {
138     /* Main Timer (1ms) */
139     uint32_t mainTmrCnt;
140     /* Timer precis (1us) */
141     bool TmrTickFlag;
142     uint32_t TmrCnt;
143     /* Measure todo flag */
144     unsigned long TmrMeas;
145     unsigned long ltime;
146     bool measTodoFlag;
147     /* Timer display */
```

```
148     uint32_t TmrDisplay;
149     /* Tmr wait shutdown */
150     bool flagCountBtnPressed;
151     uint32_t TmrBtnPressed;
152 }TIMER_DATA;
153 
154 // *****************************************************************************
155 // *****************************************************************************
156 // Section: Application Callback Routines
157 // *****************************************************************************
158 // *****************************************************************************
159 // *****************************************************************************
160 // *****************************************************************************
161 // Section: Application Initialization and State Machine Functions
162 // *****************************************************************************
163 // *****************************************************************************
164 
165 /*******************************************************************************
166   Function:
167     void APP_Initialize ( void )
168 
169   Summary:
170      MPLAB Harmony application initialization routine.
171 
172   Description:
173     This function initializes the Harmony application.  It places the
174     application in its initial state and prepares it to run so that its
175     APP_Tasks function can be called.
176 
177   Precondition:
178     All other system initialization routines should be called before calling
179     this routine (in "SYS_Initialize").
180 
181   Parameters:
182     None.
183 
184   Returns:
185     None.
186 
187   Example:
188     <code>
189     APP_Initialize();
190     </code>
191 
192   Remarks:
193     This routine must be called from the SYS_Initialize function.
194 */
195 
196 void APP_Initialize ( void );
197 
198 void prepareBuffer( char * buffer );
199 
200 void App_resetMeasFlag( void );
201 
202 /*******************************************************************************
203   Function:
204     void APP_Tasks ( void )
205 
206   Summary:
207     MPLAB Harmony Demo application tasks function
208 
209   Description:
210     This routine is the Harmony Demo application's tasks function.  It
211     defines the application's state machine and core logic.
212 
213   Precondition:
214     The system and application initialization ("SYS_Initialize") should be
215     called before calling this.
216 
217   Parameters:
218     None.
219 
220   Returns:
221     None.
```

```
222
223   Example:
224     <code>
225     APP_Tasks();
226     </code>
227
228   Remarks:
229     This routine must be called from SYS_Tasks() routine.
230  */
231
232 void APP_Tasks( void );
233
234
235 // Callback main timer
236 void MainTimer_callback( void );
237
238 // Callback display timer
239 void DisplayTimer_callback( void );
240
241 #endif /* _APP_H */
242
243 //DOM-IGNORE-BEGIN
244 #ifdef __cplusplus
245 }
246 #endif
247 //DOM-IGNORE-END
248
249 /*******************************************************************************
250  End of File
251  */
252
253
```