

```

1  /*****
2  MPLAB Harmony Application Header File
3
4  Company:
5      Microchip Technology Inc.
6
7  File Name:
8      app.h
9
10 Summary:
11     This header file provides prototypes and definitions for the application.
12
13 Description:
14     This header file provides function prototypes and data type definitions for
15     the application. Some of these are required by the system (such as the
16     "APP_Initialize" and "APP_Tasks" prototypes) and some of them are only used
17     internally by the application (such as the "APP_STATES" definition). Both
18     are defined here for convenience.
19 *****/
20
21 //DOM-IGNORE-BEGIN
22 /*****
23 Copyright (c) 2013-2014 released Microchip Technology Inc. All rights reserved.
24
25 Microchip licenses to you the right to use, modify, copy and distribute
26 Software only when embedded on a Microchip microcontroller or digital signal
27 controller that is integrated into your product or third party product
28 (pursuant to the sublicense terms in the accompanying license agreement).
29
30 You should refer to the license agreement accompanying this Software for
31 additional information regarding your rights and obligations.
32
33 SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND,
34 EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF
35 MERCHANTABILITY, TITLE, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE.
36 IN NO EVENT SHALL MICROCHIP OR ITS LICENSORS BE LIABLE OR OBLIGATED UNDER
37 CONTRACT, NEGLIGENCE, STRICT LIABILITY, CONTRIBUTION, BREACH OF WARRANTY, OR
38 OTHER LEGAL EQUITABLE THEORY ANY DIRECT OR INDIRECT DAMAGES OR EXPENSES
39 INCLUDING BUT NOT LIMITED TO ANY INCIDENTAL, SPECIAL, INDIRECT, PUNITIVE OR
40 CONSEQUENTIAL DAMAGES, LOST PROFITS OR LOST DATA, COST OF PROCUREMENT OF
41 SUBSTITUTE GOODS, TECHNOLOGY, SERVICES, OR ANY CLAIMS BY THIRD PARTIES
42 (INCLUDING BUT NOT LIMITED TO ANY DEFENSE THEREOF), OR OTHER SIMILAR COSTS.
43 *****/
44 //DOM-IGNORE-END
45
46
47 // Author M.Ricchieri
48
49
50 #ifndef _APP_H
51 #define _APP_H
52
53 // ****
54 // ****
55 // Section: Included Files
56 // ****
57 // ****
58
59 #include <stdint.h>
60 #include <stdbool.h>
61 #include <stddef.h>
62 #include <stdlib.h>
63 #include <stdio.h>
64 #include "system_config.h"
65 #include "system_definitions.h"
66
67 #include "imu/inv_imu_driver.h"
68 #include "imu/inv_imu_transport.h"
69 #include "Invn/EmbUtils/RingBuffer.h"
70 #include "I2C_ICM42670P_Functions.h"
71 #include "inv_imu_personnal_functions.h"
72
73 #include "usart_FIFO.h"
74

```

```

75 // DOM-IGNORE-BEGIN
76 #ifdef __cplusplus // Provide C++ Compatibility
77
78 extern "C" {
79
80 #endif
81 // DOM-IGNORE-END
82
83 // *****
84 // *****
85 // Section: Type Definitions
86 // *****
87 // *****
88
89 /*
90  * Select communication link between SmartMotion and IMU
91  */
92 #define SERIF_TYPE UI_I2C
93
94 /*
95  * Set power mode flag
96  * Set this flag to run example in low-noise mode.
97
98  * Reset this flag to run example in low-power mode.
99  * Note: low-noise mode is not available with sensor data frequencies less than 12.5Hz.
100  */
101 #define USE_LOW_NOISE_MODE 1
102
103 /*
104  * Select Fifo resolution Mode (default is low resolution mode)
105  * Low resolution mode: 16 bits data format
106  * High resolution mode: 20 bits data format
107  * Warning: Enabling High Res mode will force FSR to 16g and 2000dps
108  */
109 #define USE_HIGH_RES_MODE 0
110
111 /*
112  * Select to use FIFO or to read data from registers
113  */
114 #define USE_FIFO 0
115
116 /*
117  * Print raw data or scaled data
118  * 0 : print raw accel, gyro and temp data
119  * 1 : print scaled accel, gyro and temp data in g, dps and degree Celsius
120  */
121 #define SCALED_DATA_G_DPS 1
122
123 #define FAST 1
124 #define SLOW 0
125
126
127
128 typedef enum
129 {
130 APP_STATE_INIT=0,
131 APP_STATE_SERVICE,
132 APP_STATE_WAIT,
133
134 } APP_STATES;
135
136
137 typedef enum
138 {
139 SERVICE_STATE_READ_SENSORS=0,
140 SERVICE_STATE_PROCESS,
141 SERVICE_STATE_SEND_DATA_BT,
142
143 } SERVICE_STATES;
144
145
146 // Application data structure
147 typedef struct{
148

```

```
149     APP_STATES appState;
150     SERVICE_STATES serviceState;
151
152     bool isBluetoothModuleInit;
153     bool isBluetoothConnected;
154     bool isBluetoothDiscoverable;
155     bool isBluetoothInOperation;
156     bool isBluetoothInCommandMode;
157
158 } APP_DATA;
159
160
161 // Sensors data structure
162 typedef struct{
163
164     uint16_t velocity;
165     float gyroX;
166     float gyroY;
167     float gyroZ;
168     float accelX;
169     float accelY;
170     float accelZ;
171     float GyrAngleX;
172     float GyrAngleY;
173     float GyrAngleZ;
174     float batVoltage;
175     float genVoltage;
176
177 } SENS_DATA;
178
179
180 // Analogic data structure
181 typedef struct{
182
183     uint16_t AN9_V_GEN;
184     uint16_t AN10_V_BAT;
185
186 }RAW_ADC;
187
188
189
190 // *****
191 // *****
192 // Section: Application Callback Routines
193 // *****
194 // *****
195 /* These routines are called by drivers when certain events occur.
196 */
197
198 // *****
199 // *****
200 // Section: Application Initialization and State Machine Functions
201 // *****
202 // *****
203
204
205
206 // Extern variables and structures
207 extern APP_DATA      appData;
208 extern SENS_DATA     sensData;
209 extern bool          isBluetoothConnected;
210 extern bool          isBluetoothModuleInit;
211
212 extern struct inv_imu_device  myImuDevice;
213 extern struct inv_imu_serif   myImuSertif;
214
215
216 // Basic functions prototypes
217 void APP_Initialize (void);
218 void APP_Tasks( void );
219 void APP_UpdateAppState(APP_STATES NewState);
220 void clearArray(size_t arraySize, char *pArrayToClear);
221 inline void frameFormatting(char* a_dataToSend, const SENS_DATA* sensData);
222
```

```
223 // Callback functions prototypes
224 void TIMER0_Callback_Function(void);
225 void TIMER1_Callback_Function(void);
226 void TIMER5_Callback_Function(void);
227 void USART1_Callback_Function(void);
228 void imu_callback(inv_imu_sensor_event_t *event);
229
230
231 // IMU useful functions prototypes
232 int      initImuInterface(struct inv_imu_serif *icm_serif);
233 uint64_t  inv_imu_get_time_us(void);
234
235
236
237 #endif /* _APP_H */
238
239 //DOM-IGNORE-BEGIN
240 #ifdef __cplusplus
241 }
242 #endif
243 //DOM-IGNORE-END
244
245 /*****
246 End of File
247 */
```