```c
/********************************************************************************
  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 <stdio.h>
#include "system_config.h"
#include "system_definitions.h"


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

extern "C" {

#endif
// DOM-IGNORE-END

// ******************************************************************************
// ******************************************************************************
// Section: Type Definitions
// *****************************************************************************
// *****************************************************************************
// *****************************************************************************



/* Application states

  Summary:
    Application states enumeration

  Description:
    This enumeration defines the valid application states.  These states
    determine the behavior of the application at various times.
*/

typedef enum
{
    /* Application's state machine's initial state. */
    APP_STATE_INIT=0,
    APP_STATE_LINK_XBEE,
    APP_STATE_SEND_ID,
    APP_STATE_SEND_DATA,
    APP_STATE_RESET,
    APP_STATE_REFUSE,
    APP_STATE_ACCEPT,
    APP_STATE_PRESENCE,
    APP_STATE_GEST_MENU,
    APP_STATE_GET_DATA,

    /* TODO: Define states used by the application state machine. */

} APP_STATES;


// *****************************************************************************
/* Application Data

  Summary:
    Holds application data

  Description:
    This structure holds the application's data.

  Remarks:
    Application strings and buffers are be defined outside this structure.
 */

typedef struct
{
    /* The application's current state */
    APP_STATES state;

    /* TODO: Define any additional data used by the application. */

} APP_DATA;


extern char buffReadName[];// Buffer de reception de l'UART
extern uint8_t Name_Receive;// Flag de récéption
extern uint8_t countCar;// Compteur du nombre de characters d'un nom
extern uint8_t Nb_Student;
extern uint8_t Nb_Student_max;
bool flag_Com_Received;

// *****************************************************************************
// *****************************************************************************
// Section: Application Callback Routines
// *****************************************************************************
// *****************************************************************************
/* These routines are called by drivers when certain events occur.
*/

```

```c
// *****************************************************************************
// *****************************************************************************
// Section: Application Initialization and State Machine Functions
// *****************************************************************************
// *****************************************************************************

/*******************************************************************************
  Function:
    void APP_Initialize ( void )

  Summary:
     MPLAB Harmony application initialization routine.

  Description:
    This function initializes the Harmony application.  It places the
    application in its initial state and prepares it to run so that its
    APP_Tasks function can be called.

  Precondition:
    All other system initialization routines should be called before calling
    this routine (in "SYS_Initialize").

  Parameters:
    None.

  Returns:
    None.

  Example:
    <code>
    APP_Initialize();
    </code>

  Remarks:
    This routine must be called from the SYS_Initialize function.
*/

void APP_Initialize ( void );


/*******************************************************************************
  Function:
    void APP_Tasks ( void )

  Summary:
    MPLAB Harmony Demo application tasks function

  Description:
    This routine is the Harmony Demo application's tasks function.  It
    defines the application's state machine and core logic.

  Precondition:
    The system and application initialization ("SYS_Initialize") should be
    called before calling this.

  Parameters:
    None.

  Returns:
    None.

  Example:
    <code>
    APP_Tasks();
    </code>

  Remarks:
    This routine must be called from SYS_Tasks() routine.
 */

void APP_Tasks( void );

void APP_UpdateState(APP_STATES NewState);
```

```c
#endif /* _APP_H */

//DOM-IGNORE-BEGIN
#ifdef __cplusplus
}
#endif
//DOM-IGNORE-END

/*******************************************************************************
 End of File
 */
```