```c
/********************************************************************************
 *
 *    ________  _________  ____    ____  _____          ________   ______
 *   |_   __  ||  _   _  ||_   \  /   _||_   _|        |_   __  |.'  ____ \
 *     | |_ \_||_/ | | \_|  |   \/   |    | |     ______ | |_ \_|| (___ \_|
 *     |  _| _     | |      | |\  /| |    | |   _|______||   _|  _  _.____`.
 *    _| |__/ |   _| |_   _| |_\/_| |_  _| |__/ |      _| |__/ || \____) |
 *   |________|  |_____|  |_____||_____||________|      |________| \______.'
 *
 ********************************************************************************
 *
 * File          : esp.h
 * Version       : 1.0
 *
 ********************************************************************************
 *
 * Description   : Managing ESP32 state machine and commands
 *
 ********************************************************************************
 *
 * Author        : Miguel Santos
 * Date          : 25.09.2023
 *
 ********************************************************************************
 *
 * MPLAB X       : 5.45
 * XC32          : 2.50
 * Harmony       : 2.06
 *
 *******************************************************************************/

#ifndef _ESP_H
#define _ESP_H

/*******************************************************************************/

#include <stdint.h>
#include <stdbool.h>
#include "system_config.h"
#include "system_definitions.h"
#include "modules/fifo.h"
#include "modules/counter.h"

/*******************************************************************************/

/* Buffers sizes */

#define ESP_FIFO_SIZE 50
#define AT_CMD_SIZE 50
#define AT_DATA_SIZE 50
#define AT_ACK_SIZE 10

/*******************************************************************************/

/* AT commands to send */
#define AT_CMD_AT        "AT"
#define AT_CMD_RST       "AT+RST"
#define AT_CMD_CWMODEIS "AT+CWMODE=1"
#define AT_CMD_CWMODE    "AT+CWMODE?"
#define AT_CMD_CWJAP     "AT+CWJAP=\"ES-SLO-2\",\"slo-etml-es\""

/* AT acknowledge responses */
#define AT_ACK_OK        "OK"
#define AT_ACK_ERROR     "ERROR"

/*******************************************************************************/

/* Structure of packets communication as defined by ESP32 datasheet */
typedef struct
{
    char command[AT_CMD_SIZE];
    char data[AT_DATA_SIZE];
    char ack[AT_ACK_SIZE];

```

```
74  } S_AT_PACKET;
75
76  /*****************************************************************************/
77
78  /* ESP state machine */
79  typedef enum
80  {
81      /* Waiting for a command */
82      ESP_STATE_IDLE,
83
84      /* Transmitting a command */
85      ESP_STATE_TRANSMIT,
86
87      /* Receiving a command */
88      ESP_STATE_RECEIVE,
89
90      /* Translating the UART message */
91      ESP_STATE_TRANSLATE,
92
93      /* Waiting for main application */
94      ESP_STATE_WAIT,
95
96  } E_ESP_STATES;
97
98  /*****************************************************************************/
99
100 /* ESP32 structure of global data */
101 typedef struct
102 {
103     /* Application's current states */
104     E_ESP_STATES state;
105
106     /* Applications's flags */
107     bool transmit;
108     bool receive;
109     bool translate;
110     bool wait;
111     bool newMessage;
112
113     /* Applications COUNTERS */
114     S_Counter cntReceive;
115     S_Counter cntWait;
116
117     /* Application's FIFOS descriptors */
118     S_Fifo fifoDesc_tx;
119     S_Fifo fifoDesc_rx;
120
121     /* Application's FIFOS buffers */
122     uint8_t fifoBuff_tx[ESP_FIFO_SIZE];
123     uint8_t fifoBuff_rx[ESP_FIFO_SIZE];
124
125     /* Buffer to store response commands */
126     char resBuffer[ESP_FIFO_SIZE];
127
128     /* Pointer to last char received */
129     char *p_resBuffer;
130
131     /* Store response as different fields */
132     S_AT_PACKET atResponse;
133
134 } ESP_DATA;
135
136 /*****************************************************************************/
137
138 /**
139  * @brief ESP_Initialize
140  *
141  * Initialize ESP32 state machine, counters and FIFOs
142  *
143  * @param  void
144  * @return void
145  */
146 void ESP_Initialize ( void );
```

```
147
148 /*****************************************************************************/
149
150 /**
151  * @brief ESP_Tasks
152  *
153  * Execute ESP32 state machine, should be called cyclically
154  *
155  * @param  void
156  * @return void
157  */
158 void ESP_Tasks( void );
159
160 /*****************************************************************************/
161
162 /**
163  * @brief ESP_SendCommand
164  *
165  * Send a command to the ESP32, managed by state machine
166  *
167  * @param  char Command to send ; Use constant definitions
168  * @return bool True = command send ; False = Not allowed to send a command
169  */
170 bool ESP_SendCommand( char *command );
171
172 /*****************************************************************************/
173
174 #endif /* _ESP_H */
175
176 /* End of File ***************************************************************/
177
```