```
/* ********************************************************************** */
/* File name:        DroneWiFi.h                                          */
/* File description: WiFi socket connection handling  header file         */
/* Author name:      Giacomo Dollevedo, Gustavo Fernandes                 */
/* Creation date:    18nov2020                                            */
/* Revision date:    04jan2021                                            */
/* ********************************************************************** */

#ifndef DroneWiFi_h
#define DroneWiFi_h

#include <WiFi.h>
#include <WiFiMulti.h>

//#include <Arduino.h>

#define IP_GIA "192.168.43.182"
#define PORT 25565
#define SSID_GIA "Gio"
#define PASS_GIA "nematodeo"
#define SSID_GUS "Guslifer-Wifi"
#define PASS_GUS "12345678"

#define VEL_INC 5
#define THROTTLE_INC 10

typedef struct droneParams
 {
  int M1, M2, M3, M4;
  float H, Battery;
  int setPoint;
  int time;

 };

typedef struct rotVel
 {
  int roll = 0;
  int pitch = 0;
  int yaw = 0;
  int throttle = 0;

 };

 typedef struct pidGains
 {
  float kp = 0;
  float ki = 0;
  float kd = 0;
 };

class DroneWiFi
{
public:

/*
**********************************************************************************
*/
```

```
/* Method's name:          initWiFi
   */
/* Description:            WiFi initialization. Connects to a network and server.
   */
/*                         Authenticates.
   */
/*
    */
/* Entry parameters:       char* ssid -> WiFi network ssid
    */
/*                         char* pass -> WiFi network password
    */
/*                         char* hostIP -> Server connection IP
   */
/*                         int port    -> Server Connection port
    */
/*
    */
/* Return parameters:      n/a
    */
/*
*************************************************************************
*/
 void initWiFi(char* ssid, char* pass, char* hostIP, int port);


/*
*************************************************************************
*/
/* Method's name:          sendData
   */
/* Description:            WiFi comm. method to send data. Will mark the first 5
bytes  */
/*                         with the message length for the receiver end.
    */
/*
    */
/* Entry parameters:       WiFiClient _serverCon -> Socket object that represents the
   */
/*                                      connection
    */
/*                         String message -> Message that will be sent through socket
   */
/*                         char* len_c -> Array to enable the first 5 bytes marker
    */
/*
    */
/* Return parameters:      n/a
    */
/*
*************************************************************************
*/
 void sendData(String message);


/*
*************************************************************************
*/
```

```
/* Method's name:          receiveData
   */
/* Description:            WiFi comm. method to receive data. Will read the first 5
  */
/*                         incoming bytes and use it to read the incoming stream
   */
/*
   */
/* Entry parameters:       WiFiClient _serverCon -> Socket object that represents the
  */
/*                                          connection
   */
/*
   */
/* Return parameters:      String -> Message read from the connection
  */
/*
*************************************************************************************
*/
 String receiveData();


/*
*************************************************************************************
*/
/* Method's name:          setParams
   */
/* Description:            Set internal droneParams variable
   */
/*
   */
/* Entry parameters:       droneParams prm -> struct to be set
   */
/*
   */
/* Return parameters:      n/a
   */
/*
*************************************************************************************
*/
 void setParams(droneParams prm);


/*
*************************************************************************************
*/
/* Method's name:          processComm
   */
/* Description:            State machine to process server communication commands
  */
/*
   */
/* Entry parameters:       String msg -> server command to be handled
  */
/*
   */
/* Return parameters:      n/a
   */
```

```
/*
*********************************************************************************
*/
 void processComm(String msg);


/*
*********************************************************************************
*/
/* Method's name:          getParams
   */
/* Description:            Return the internal drone parameters struct
   */
/*
   */
/* Entry parameters:       n/a
   */
/*
   */
/* Return parameters:      droneParams -> internal parameters struct
   */
/*
*********************************************************************************
*/
 droneParams getParams();


/*
*********************************************************************************
*/
/* Method's name:          getVel
  */
/* Description:            Return the internal joystick setpoints struct
   */
/*
   */
/* Entry parameters:       n/a
   */
/*
   */
/* Return parameters:      rotVel -> internal joystick setpoints struct
  */
/*
*********************************************************************************
*/
 rotVel getVel();

/*
*********************************************************************************
*/
/* Method's name:          getPIDGains
   */
/* Description:            Return the internal drone PID gains to specific axis
  */
/*
   */
/* Entry parameters:       unsigned char axis -> desired axis gains
  */
```

```
/*
   */
/* Return parameters:      pidGains -> internal PID gains struct
   */
/*
*************************************************************************************
*/
 pidGains getPIDGains(unsigned char axis);


/*
*************************************************************************************
*/
/* Method's name:          enableDebug
   */
/* Description:            Enables Serial Comm Printing
  */
/*
   */
/* Entry parameters:       n/a
   */
/*
   */
/* Return parameters:      n/a
   */
/*
*************************************************************************************
*/
 void enableDebug();


/*
*************************************************************************************
*/
/* Method's name:          disableDebug
*/
/* Description:            Disables Serial Comm Printing
   */
/*
   */
/* Entry parameters:       n/a
   */
/*
   */
/* Return parameters:      n/a
   */
/*
*************************************************************************************
*/
 void disableDebug();


private:


  /*Object to handle network connection*/
  WiFiMulti _wifiCon;

  /*Object to handle server connection*/
```

```
183   WiFiClient _serverCon;
184
185   /*Struct to monitor drone parameters*/
186   droneParams _params;
187
188   /*Struct to monitor joystick setpoints*/
189   rotVel  _joystickSetpoints;
190
191   /*PID Gains struct for each axis*/
192   pidGains _pidRoll;
193   pidGains _pidPitch;
194   pidGains _pidYaw;
195
196 /*
    ************************************************************************
    */
197 /* Method's name:           connectWifi
       */
198 /* Description:             Attempts wifi network connection
      */
199 /*
       */
200 /* Entry parameters:        char* ssid -> Network SSID (name)
       */
201 /*                          char* pass -> Network password
      */
202 /*
       */
203 /* Return parameters:       unsigned char -> 0 == failed to connect / 1 == success
      */
204 /*
    ************************************************************************
    */
205   unsigned char connectWifi(char* ssid, char* pass);
206
207 /*
    ************************************************************************
    */
208 /* Method's name:           connectServer
       */
209 /* Description:             Attempts server connection through WiFi network
       */
210 /*
       */
211 /* Entry parameters:        char* hostIP -> Server connection IP
      */
212 /*                          int port     -> Server Connection port
       */
213 /*
       */
214 /* Return parameters:       unsigned char -> 0 == failed to connect / 1 == success
      */
215 /*
    ************************************************************************
    */
216   unsigned char connectServer(char* hostIP, int port);
217
218
219   /*Internal Variable to create the length stamps on messages*/
```

```
  char* _len_marker = (char*)calloc(5, sizeof(char));

  /*Internal flag to joystick status*/
  unsigned char joystick_enabled = 0;

  /*Internal flag to serial comm status*/
  unsigned char debugging_enabled = 1;

};

#endif

```