

```

1  /*****
2  * Descrição: Arquivo header para implementação das funções de
3  * interface de controle de velocidade dos motores do Drone
4  * Autores: Gustavo L. Fernandes e Giácomo A. Dollevedo
5  * Ultima Atualização: 03/01/2021
6  *****/
7
8  #ifndef ThrottleControl_h
9  #define ThrottleControl_h
10
11 #include "FlightControl.h"
12 #include "Arduino.h"
13 #include <analogWrite.h>
14 #include <ESP32PWM.h>
15 #include <ESP32Servo.h>
16 #include <ESP32Tone.h>
17
18 #define MAXTHROTTLE 1600
19 #define MOTORTHROTTLE 1060 //sinal em que os motores começam a atuar
20 #define PINMOTOR1 13
21 #define PINMOTOR2 27
22 #define PINMOTOR3 14
23 #define PINMOTOR4 12
24
25
26
27
28 class ThrottleControl
29 {
30
31     private:
32
33     Servo _m1;
34     Servo _m2;
35     Servo _m3;
36     Servo _m4;
37     int _throttle;
38     int* _actualVel;
39
40     public:
41     /* *****/
42     /* Nome do metodo:          ThrottleControl
43     /*
44     /* Descrição:              Método construtor do objeto, não faz nada além de
45     /*                          associar 4 objetos do tipo Servo que serão utilizados
46     /*
47     /* Parametros de entrada:   Nenhum (Vazio)
48     /*
49     /* Parametros de saída:     Nenhum (Vazio)
50     /*
51     /*

```

```

50  /*
    */
51  /* *****
    */
52      ThrottleControl();
53
54  /* *****
    */
55  /* Nome do metodo:          initializeMotors
    */
56  /* Descrição:              Inicia cada um dos motores criados, atrelando a uma
saída*/
57  /*                          PWM do ESP32 e setando os limites max e min que define
    */
58  /*                          também a inicialização dos 4 ESC
    */
59  /*
    */
60  /* Parametros de entrada: int pinMotor1, int pinMotor2, int pinMotor3, int
pinMotor4*/
61  /*                          Pinos PWM que serão conectados os sinais de controle
    */
62  /* Parametros de saída: Nenhum (Vazio)
    */
63  /*
    */
64  /*
    */
65  /* *****
    */
66
67      void initializeMotors(int pinMotor1, int pinMotor2, int pinMotor3, int
pinMotor4);
68
69  /* *****
    */
70  /* Nome do metodo:          setActualVel
    */
71  /* Descrição:              Define as velocidades que serão atingidas pelos 4
motores*/
72  /*
    */
73  /*
    */
74  /* Parametros de entrada: int desiredVel1, int desiredVel2, int desiredVel3,
    */
75  /*                          int desiredVel4 , que são as velocidades desejadas
    */
76  /*                          aceita valores entre 1000 e 2000 mas limita em 1500
    */
77  /* Parametros de saída: Nenhum (Vazio)
    */
78  /*
    */
79  /*
    */
80  /* *****
    */

```

```

81         void setActualVel(int desiredVel1, int desiredVel2, int desiredVel3, int
desiredVel4);
82     /* *****
83     /* Nome do metodo:           getActualVel
84     /* Descrição:               Consulta o sinal pwm definido nos 4 motores
85     /*
86     /*
87     /* Parametros de entrada: Nenhum (Vazio)
88     /*
89     /* Parametros de saida:      int* array com os 4 valores de velocidades lidos
90     /*
91     /*
92     /* *****
93     int* getActualVel();
94
95     /* *****
96     /* Nome do metodo:           gtestMotors
97     /* Descrição:               Testa se os motores estão funcinoando e respondendo
98     /*
99     /*
100    /* Parametros de entrada: Nenhum (Vazio)
101    /*
102    /* Parametros de saida:      boolean   (1 Funcionando) (0 Com problemas)
103    /*
104    /*
105    /* *****
106    void testMotors();
107
108    /* *****
109    /* Nome do metodo:           Control
110    /* Descrição:               Distribua a velocidade controlada para os 4 motores
111    /*
112    /*

```

```

112  /*
113  /* Parametros de entrada: FlightControl pidRoll, FlightControl pidPitch,
114  /*          IMU imu , que são os objetos do controlador
115  /*          implementado para cada eixo e sensor de inercia
116  /*
117  /* Parametros de saida: Vazio (Nenhum)
118  /*
119  /*
120  /* *****
121  void Control(FlightControl pidRoll, FlightControl pidPitch,FlightControl pidYaw,
float rollVel, float pitchVel, float yawVel);
122
123  /* *****
124  /* Nome do metodo:          SingleAxisVelControl
125  /* Descrição:              Distribui a velocidade controlada para os 2 motores
126  /*                          de modo a contrlar apenas um eixo de movimento
127  /*
128  /* Parametros de entrada: FlightControl pidRoll que é objeto do eixo roll de
129  /*                          controle ,
130  /*
131  /*
132  /* Parametros de saida: Vazio (Nenhum)
133  /*
134  /*
135  /* *****
136  void SingleAxisVelPitchControl(FlightControl pidPitch);
137
138
139  /* *****
140  /* Nome do metodo:          SingleAxisVelControl
141  /* Descrição:              Distribui a velocidade controlada para os 2 motores
142  /*                          de modo a contrlar apenas um eixo de movimento
143  /*

```

```
143  /*
    */
144  /* Parametros de entrada: FlightControl pidRoll que é objeto do eixo roll de
    */
145  /*           controle           ,
    */
146  /*
    */
147  /*
    */
148  /* Parametros de saida:  Vazio (Nenhum)
    */
149  /*
    */
150  /*
    */
151  /* *****
    */
152      void SingleAxisVelRollControl(FlightControl pidRoll);
153
154  /* *****
    */
155  /* Nome do metodo:           getThrottle
    */
156  /* Descrição:                Consulta a velocidade (pwm) base atual dos motores
    */
157  /*
    */
158  /*
    */
159  /* Parametros de entrada: Vazio (Nenhum)
    */
160  /*
    */
161  /*
    */
162  /*
    */
163  /* Parametros de saida:  _throttle (int)
    */
164  /*
    */
165  /*
    */
166  /* *****
    */
167      int getThrottle();
168
169  /* *****
    */
170  /* Nome do metodo:           setThrottle
    */
171  /* Descrição:                Define nova velocidade (pwm) base atual dos motores
    */
172  /*
    */
173  /*
    */
```

```
174 /* Parametros de entrada: throttleDesired (int)
    */
175 /*
    */
176 /*
    */
177 /*
    */
178 /* Parametros de saida:  Vazio (Nenhum )
    */
179 /*
    */
180 /*
    */
181 /* *****
    */
182
183     void setThrottle(int throttleDesired);
184
185 };
186
187 #endif
188
```