```
/*****************************************************
 * Descrição: Arquivo header das funções responsáveis
 * pelo controle de voo do drone.
 * Autores: Gustavo L. Fernandes e Giácomo A. Dollevedo
 * Ultima Atualização: 6/01/2021
*****************************************************/
#ifndef FlightControl_h
#define FlightControl_h

#include "Arduino.h"
#include "IMU.h"
#include <analogWrite.h>
#include <ESP32PWM.h>
#include <ESP32Servo.h>
#include <ESP32Tone.h>

#define PIDMAX 400

typedef struct kPID
{
float fkp, fki, fkd;
};

class FlightControl
{

  private:
    kPID _gains;
    float _fsetPoint;
    float _fpidCalculated;
    float _fimem;
    float _ferrorPrevious;
    char  _axis;
    float _setPoint;


  public:
/* ************************************************************************************
*/
/* Nome do metodo:          FlighControl  - Construtor Vazio
*/
/* Descrição:               Inicializa um objeto controlador padrão, ou seja
*/
/*                          com valores de ganho nulos
*/
/*                          Setpoint nulo e constante
*/
/*
*/
/* Parametros de entrada:  Nenhum (Vazio)
*/
/*
*/
/*
*/
/* Parametros de saida: Nenhum (Vazio)
*/
```

```
/*
*/
/*
*/
/* ************************************************************************
*/
    FlightControl(char Axis);

/* ************************************************************************
*/
/* Nome do metodo:          FlighControl  - Construtor Pre Definido
*/
/* Descrição:               Inicializa um objeto controlador com ganhos definidos
*/
/*
*/
/*
*/
/* Parametros de entrada: (float fkp, float fki, float fkd) ganhos dos
controladores*/
/*                        proporcional, integrador e derivativo
*/
/*                         Setpoint nulo e constante
*/
/*                        (char Axis) podendo ser "r", "p", "y" para determinar
*/
/*                        qual o eixo que o objeto será definido
*/
/*
*/
/* Parametros de saida: Nenhum (Vazio)
*/
/*
*/
/*
*/
/* ************************************************************************
*/
    FlightControl(float fkp, float fki, float fkd, char Axis);

/* ************************************************************************
*/
/* Nome do metodo:          pidVelControl
*/
/* Descrição:                - Rotina de Controle de velocidade
*/
/*                          para um determinado eixo
*/
/*
*/
/*
*/
/* Parametros de entrada: (float axisVel) variavel que contem os valores medidos
*/
/*  de velocidade lidos da IMU que será dado como entrada pro controlador
*/
/* Parametros de saida: Nenhum (Vazio)
*/
```

```
/*
*/
/*
*/
/* ************************************************************************
*/
    void pidVelControl(float  axisVel);

/* ************************************************************************
*/
/* Nome do metodo:          getPID_Calculated
*/
/* Descrição:                - retorna o valor de PID calculado a ser utilizado
*/
/*                            para saida do sistema
*/
/*
*/
/* Parametros de entrada: Nenhum (Vazio)
*/
/*
*/
/* Parametros de saida: float _pidCalculated
*/
/*
*/
/*
*/
/* ************************************************************************
*/
    float getPID_Calculated();

/* ************************************************************************
*/
/* Nome do metodo:          getGains
*/
/* Descrição:                - retorna o valor dos ganhos do controlador PID
*/
/*
*/
/*
*/
/* Parametros de entrada: Nenhum (Vazio)
*/
/*
*/
/* Parametros de saida: kPID _gains
*/
/*
*/
/*
*/
/* ************************************************************************
*/
    kPID getGains();

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

```
/* Nome do metodo:          setKp
*/
/* Descrição:                - Seta um novo valor para o ganho do controlador
*/
/*                             proporcional
*/
/*
*/
/* Parametros de entrada: float kp
*/
/*
*/
/* Parametros de saida: Nenhum (Vazio)
*/
/*
*/
/*
*/
/* ************************************************************************
*/

    void setKp(float kp);

/* ************************************************************************
*/
/* Nome do metodo:          setKd
*/
/* Descrição:                - Seta um novo valor para o ganho do controlador
*/
/*                             derivativo
*/
/*
*/
/* Parametros de entrada: float kd
*/
/*
*/
/* Parametros de saida: Nenhum (Vazio)
*/
/*
*/
/*
*/
/* ************************************************************************
*/
    void setKd(float kd);

/* ************************************************************************
*/
/* Nome do metodo:          setKi
*/
/* Descrição:                - Seta um novo valor para o ganho do controlador
*/
/*                             integrativo
*/
/*
*/
/* Parametros de entrada: float ki
*/
```

```
/*
*/
/* Parametros de saida: Nenhum (Vazio)
*/
/*
*/
/*
*/
/* ************************************************************************
*/
    void setKi(float ki);

/* ************************************************************************
*/
/* Nome do metodo:          setSetPoint
*/
/* Descrição:                - Seta um novo valor para o setPoint do controlador
*/
/*                             de velocidade
*/
/*
*/
/* Parametros de entrada: float newSetPoint
*/
/*
*/
/* Parametros de saida: Nenhum (Vazio)
*/
/*
*/
/*
*/
/* ************************************************************************
*/
    void setSetPoint(float newSetPoint);

/* ************************************************************************
*/
/* Nome do metodo:          getSetPoint
*/
/* Descrição:                - Consulta o atual valor para o setPoint do
controlador*/
/*
*/
/*
*/
/* Parametros de entrada: Nenhum (Vazio)
*/
/*
*/
/* Parametros de saida: float _setPoint
*/
/*
*/
/*
*/
/* ************************************************************************
*/
    float getSetPoint();
```

```
175
176 /* ************************************************************************
    */
177 /* Nome do metodo:          getAccError
    */
178 /* Descrição:                - Consulta o erro acumulado do controlador
    */
179 /*
    */
180 /*
    */
181 /* Parametros de entrada: Nenhum (Vazio)
    */
182 /*
    */
183 /* Parametros de saida: float _fimem
    */
184 /*
    */
185 /*
    */
186 /* ************************************************************************
    */
187
188     float getAccError();
189
190 /* ************************************************************************
    */
191 /* Nome do metodo:          getPreviousError
     */
192 /* Descrição:                - Consulta o erro previo  do controlador
    */
193 /*
    */
194 /*
    */
195 /* Parametros de entrada: Nenhum (Vazio)
    */
196 /*
    */
197 /* Parametros de saida: float _ferrorPrevious
    */
198 /*
    */
199 /*
    */
200 /* ************************************************************************
    */
201
202     float getPreviousError();
203
204 };
205
206
207
208
209
210 #endif
211
```