



COMPETIÇÃO DE ROBÔS VOADORES INTELIGENTES

Manual API – Drone

João L. Vilaça



Índice

1.	Sistema de eixo e movimentos possíveis do Drone	3
2.	Lista de funções para controlo dos movimentos do drone	3
2.1	Ligação PC <-> Drone.....	3
2.2	Calibrar do Drone	4
2.3	Movimentar Drone.....	4
2.4	Ler estado do Drone.....	6
2.5	Manipulação das câmaras do Drone.....	8
2.6	Acesso ao módulo de GPS.....	8
2.7	Outras funções	8

1. Sistema de eixo e movimentos possíveis do Drone



2. Lista de funções para controlo dos movimentos do drone

2.1 Ligação PC <-> Drone

/// *Estabelece a ligação entre o PC e o Drone.*

/// *Esta função deve ser utilizada antes de qualquer outra função.*

void droneLigar()

void iDroneCup_Connect()

/// Desliga a ligação entre o PC e o Drone.

void droneDesligar()

void iDroneCup_Disconnect()

/// Remove a ligação entre o PC e o Drone. Permite que outro PC efetue uma ligação ao Drone.

void droneRemoverLigação()

2.2 Calibrar do Drone

/// Calibra os acelerómetros do Drone.

/// Permite definir um novo plano de orientação do Drone.

void droneCalibrar()

void iDroneCup_Calibration()

2.3 Movimentar Drone

/// Faz descolar e estabilizar o Drone a uma altura de 1 metro

void droneDescolar()

void iDroneCup_TakeOff()

/// Aterra suavemente o Drone na posição actual.

void droneAterrar()

/// Desliga os motores e efectua uma aterragem brusca. Deve ser apenas chamada em situações de emergência! Pode danificar o Drone!

void droneEmergencia()

/// Reactiva o Drone após uma aterragem de emergência.

void droneResetEmergencia()

/// Indica ao Drone que deve parar o movimento actual e pairar sobre a posição actual.

```
void dronePairar()
```

```
void iDroneCup_Hover()
```

// Este comando faz o Drone mover-se para a frente. (direcção +x)

/// <param name="velocidade">Valor entre [0...1].</param> 0 velocidade mínima, 1 velocidade máxima.

```
void droneAvancar(float velocidade)
```

/// Este comando faz o Drone mover-se para a trás. (direcção -x)

/// <param name="velocidade">Valor entre [0...1].</param> 0 velocidade mínima, 1 velocidade máxima.

```
void droneRecuar(float velocidade)
```

/// Este comando faz o Drone mover-se para a esquerda. (direcção -y)

/// <param name="velocidade">Valor entre [0...1].</param> 0 velocidade mínima, 1 velocidade máxima.

```
void droneEsquerda(float velocidade)
```

/// Este comando faz o Drone mover-se para a direita. (direcção +y)

/// <param name="velocidade">Valor entre [0...1].</param> 0 velocidade mínima, 1 velocidade máxima.

```
void droneDireita(float velocidade)
```

/// Este comando faz o Drone mover-se para a cima. (direcção +z)

/// <param name="velocidade">Valor entre [0...1].</param> 0 velocidade mínima, 1 velocidade máxima.

```
void droneSubir(float velocidade)
```

/// Este comando faz o Drone mover-se para a baixo. (direcção -z)

/// <param name="velocidade">Valor entre [0...1].</param> 0 velocidade mínima, 1 velocidade máxima.

```
public void droneDescer(float velocidade)
```

```
///  
// Este comando faz o Drone rodar para a direita. (rotação +rot_z)  
// <param name="velocidade">Valor entre [0...1].</param>0 velocidade mínima, 1  
// velocidade máxima.  
void droneRodarDireita(float velocidade)
```

```
///  
// Este comando faz o Drone rodar para a esquerda. (rotação -rot_z)  
// <param name="velocidade">Valor entre [0...1].</param>0 velocidade mínima, 1  
// velocidade máxima.  
void droneRodarEsquerda(float velocidade)
```

```
///  
// Este comando efectua movimentos simultâneos nos 3 eixos (X,Y,Z) e rotação em Z.  
// <param name="velocidadeX">Valor entre [-1...0...1].</param> (eixo x) -1 recua à  
// velocidade máxima; 1 frente à velocidade máxima; 0 não tem movimento em x  
// <param name="velocidadeY">Valor entre [-1...0...1].</param>(eixo y) -1 esquerda à  
// velocidade máxima; 1 direita à velocidade máxima; 0 não tem movimento em y  
// <param name="velocidadeZ">Valor entre [-1...0...1].</param>(eixo z) -1 subir à  
// velocidade máxima; 1 descer à velocidade máxima; 0 não tem movimento em z  
// <param name="velocidadeROT">Valor entre [-1...0...1].</param>(eixo x) -1 rodar à direita  
// à velocidade máxima; 1 rodar à esquerda à velocidade máxima; 0 não tem rotação em z  
void droneMoverPRO(float velocidadeX, float velocidadeY, float velocidadeZ, float velocidade  
ROT)
```

2.4 Ler estado do Drone

```
///  
// Indica a carga actual da bateria.  
int droneObterValorBateria()
```

```
///  
// Indica a versão de firmware .  
String droneObterVersion()
```

/// Indica o valor actual de pitch.
double droneObterPitch()

/// Indica o valor actual de Roll.
float droneObterRoll()

/// Indica o valor actual de Yaw.
float droneObterYaw()

/// Indica o valor actual da altitude
float droneObterAltitude()

/// Indica a velocidade actual em x
float droneObterVelocidadeX()

/// Indica a velocidade actual em y
float droneObterVelocidadeY()

/// Indica a velocidade actual em z
float droneObterVelocidadeZ()

/// Indica se o drone já aterrou
bool iDroneCup_isLanded()

/// Indica se o drone está a pairar
bool iDroneCup_isHovering()

/// Indica se o drone está a voar
bool iDroneCup_isFlying()

/// Indica se o drone está a aterrar
bool iDroneCup_isLanding()

2.5 Manipulação das câmaras do Drone

```
/// Muda a câmara  
/// <param name=" id ">Valor 0 ou 1</param>  
void droneMudarCamara(DroneCamera id)
```

```
/// devolve um frame da câmara seleccionada.  
Bitmap droneObterImagem()
```

2.6 Acesso ao módulo de GPS

```
bool iDroneCup_isGpsReady()
```

```
void iDroneCup_Read_Heading()
```

```
bool iDroneCup_isCompassReady()
```

2.7 Outras funções

```
void iDroneCup_Set_MaximumHeight()
```

```
void iDroneCup_RunAction(...)
```

```
void iDroneCup_RunLeds(...)
```