

AR Drone Control API Notes

The drone can be controlled using AT commands. Please refer to the SDK manual for specifics on using the AT commands and general control guidelines. However, please use these notes for specific command parameters. We have found some errors in the SDK.

AT commands are text strings sent to the drone to control its actions through UDP. The drone's UDP port is 5556 (there other ports for the NavData and camera feeds) According to tests, effective control of the drone is reached by sending the AT commands every 30 ms for smooth drone movements. To prevent the drone from considering the WIFI connection as lost, two consecutive commands must be sent within less than 2 seconds. In addition, to prevent the drone from setting the watchdog bit, it is recommended that something is always being sent every 50-100ms. Since UDP is an unreliable datagram service, more frequent transmissions may help prevent the drone from setting the watchdog bit in case of dropped or corrupted packets.

AT Command Syntax

Strings are encoded as 8-bit ASCII characters, with a Line Feed character (byte value 10(10)), noted as <LF>. In C this is “\r.” A single UDP packet can contain one or more commands, separated by a line feed. An AT command must reside in a single UDP packet. Splitting an AT command in two or more UDP packets is not possible.

Example : AT*PCMD=21625,1,0,0,0,0<LF>AT*REF=21626,290717696<LF>

The maximum length of the total command cannot exceed 1024 characters; otherwise the entire command line is rejected. This limit is hard coded in the drone software.

Most commands will accept arguments, which can be of three different types:

- A signed integer, stored in the command string with a decimal representation (ex: the sequence number)
- A string value stored between double quotes (ex: the arguments of AT*CONFIG)
- A single-precision IEEE-754 floating-point value (aka. float). Those are never directly stored in the command string. Instead, the 32-bit word containing the float will be considered as a 32-bit signed integer and printed in the AT command

Here are some of the commands:

AT*REF=<sequence>,<UI>

AT*PCMD=<sequence>,<enable>,<pitch>,<roll>,<gaz>,<yaw>

AT*CONFIG=<sequence>,\"<name>\", \"<value>\"

Examples (using C syntax)

```
void droneTakeOff(void) {  
    // Take off  
    snprintf(droneBuf, MAX_COMMAND_LEN, "AT*REF=%u,290718208", droneSeq);  
    //droneSeq is the sequence number and should be incremented each time  
}
```

```
void droneLand(void) {  
    // Land the drone  
    snprintf(droneBuf, MAX_COMMAND_LEN, "AT*REF=%u,290717696", droneSeq);  
}
```

```
void droneMove(void) {  
    // Have the drone hover  
    snprintf(droneBuf, MAX_COMMAND_LEN, "AT*PCMD=%u,1,%d,%d,%d,%d",  
droneSeq, pitch, roll, gaz, yaw);  
}
```

```
void droneHover(void) {  
    // Have the drone hover  
    snprintf(droneBuf, MAX_COMMAND_LEN, "AT*PCMD=%u,1,0,0,0,0", droneSeq);  
}
```

```
void droneUp(void) {  
    // Move drone up, same as gaze  
    snprintf(droneBuf, MAX_COMMAND_LEN, "AT*PCMD=%u,1,0,0,1045220557,0",  
droneSeq);  
}
```

```
void droneDown(void) {  
    // Move drone down  
    snprintf(droneBuf, MAX_COMMAND_LEN, "AT*PCMD=%u,1,0,0,-1102263091,0",  
droneSeq);  
}
```

```
void droneForward(void) {  
    // Move drone forward, same as pitch
```

```

        snprintf(droneBuf, MAX_COMMAND_LEN, "AT*PCMD=%u,1,0,-1102263091,0,0",
droneSeq);
    }

void droneBack(void) {
    // Move drone back
    snprintf(droneBuf, MAX_COMMAND_LEN, "AT*PCMD=%u,1,0,1045220557,0,0",
droneSeq);
}

void droneLeft(void) {
    // Move drone left, same as roll
    snprintf(droneBuf, MAX_COMMAND_LEN, "AT*PCMD=%u,1,-1102263091,0,0,0",
droneSeq);
}

void droneRight(void) {
    // Move drone right
    snprintf(droneBuf, MAX_COMMAND_LEN, "AT*PCMD=%u,1,1045220557,0,0,0",
droneSeq);
}

void droneRotateLeft(void) {
    // Rotate drone left, same as yaw
    snprintf(droneBuf, MAX_COMMAND_LEN, "AT*PCMD=%u,1,0,0,0,-1085485875",
droneSeq);
}

void droneRotateRight(void) {
    // Rotate drone right
    snprintf(droneBuf, MAX_COMMAND_LEN, "AT*PCMD=%u,1,0,0,0,1061997773",
droneSeq);
}

```

Example Float to Integer conversions:

(float)0.05 = (int)1028443341	(float)-0.05 = (int)-1119040307
(float)0.1 = (int)1036831949	(float)-0.1 = (int)-1110651699
(float)0.2 = (int)1045220557	(float)-0.2 = (int)-1102263091
(float)0.5 = (int)1056964608	(float)-0.5 = (int)-1090519040

Other useful commands:

emergency reset: AT*REF=<sequence>,290717952

set max altitude (3 meter): AT*CONFIG=1,\"control:altitude_max\", \"3000\"

set flat trim: AT*FTRIM=<sequence>

reset watchdog bit: AT*COMWDG=<sequence>