

# HawkEye 6300 Serial API

---

RS232/RS485 Serial control API for the HawkEye 6300 and YB3i devices

Last update: 5th September 2016

# Overview

The serial command API enables 3rd party control systems to interact with the HawkEye 6300.

They can send and receive their own data, and control elements of the HawkEye 6300's configuration. It is designed to be deliberately restrictive, with the overriding principle being that the external control system should not compromise the autonomous tracking capability of the device.

# Table of contents

<b>Overview</b>	<b>2</b>
<b>Table of contents</b>	<b>3</b>
<b>Change Log</b>	<b>4</b>
<b>Link specification</b>	<b>5</b>
<b>Commands</b>	<b>6</b>
Request GPS position	6
GPS Hot mode	6
Tracking frequency	7
Transmit data	8
Checking message status	8
Receive data	9

# Change Log

## Date

---

28/October/2014	Initial Version
19/December/2014	Updated to reflect the HawkEye 6300 release 1.3.1
10/June/2015	Inclusion of MT messaging commands
5/September/2016	Corrected line terminations

# Link specification

The serial API is available as either an RS232 or an RS485 interface. In either case, the recommended communication settings are 8 data bits, no parity bit, and 1 stop bit (8-N-1).

It is possible to configure the baud rate, using the app. We recommend 19200 baud.

RS232 versions are '3-wire' with RX, TX and GND signals.

RS485 versions are opto-isolated, with A, B, Y, Z and GND signals.

There is no flow-control.

# Commands

All communication is performed as command/response pairs. The 3rd party controller always issues commands, and the HawkEye 6300 always issues responses.

Commands and responses are always terminated with a carriage-return character (\r).

## Request GPS position

This command returns the current GPS position of the device. Note that the GPS must be switched on for this to work (see GPS Hot mode below)

```
R7+GPS\r
```

```
00117.49527,W,5053.33916,N,,1.001,20.7,3,06,3.31,114401.00,191214\r  
OK\r
```

<latitude>,<EW>,<longitude>,<NS>,<course>,<speed>,<altitude>,<quality>,  
<satellites>,<hdop>,<time>,<date>\r

## GPS Hot mode

Controls the HawkEye 6300 GPS module. If a position is expected, then the GPS needs to be switched on ('Hot') in sufficient time to acquire a fix.

```
R7+GPS:HOT=1\r
```

```
OK\r
```

As with all configuration commands, the current state can be queried by omitting the argument:

```
R7+GPS:HOT\r
```

```
1\r  
OK\r
```

Possible values are 0 (off) and 1 (on).

## Tracking frequency

Controls the HawkEye 6300 tracking frequency.

```
R7+TRACK:FREQ=5\r
```

```
OK\r
```

Query current tracking frequency:

```
R7+TRACK:FREQ\r
```

```
11\r
```

```
OK\r
```

### Transmit Frequency values

0	Continuous
1	5 minutes
2	10 minutes
3	15 minutes
4	20 minutes
5	30 minutes
6	60 minutes
7	90 minutes
8	120 minutes (2 hours)
9	180 minutes (3 hours)
10	240 minutes (4 hours)
11	360 minutes (6 hours)
12	480 minutes (8 hours)
13	720 minutes (12 hours)
14	Burst Mode

## Transmit data

This command instructs the tracker to send user-supplied data.

Data is hex-encoded. The HawkEye 6300 will respond with a message ID, which can be used later to query status.

The checksum is CRC-CCITT (0x0000) 16 bit, of all following bytes. There is an example calculator online here:

[http://depa.usst.edu.cn/chenjq/www2/software/crc/CRC\\_Javascript/CRCcalculation.htm](http://depa.usst.edu.cn/chenjq/www2/software/crc/CRC_Javascript/CRCcalculation.htm)

Example 'Hello world':

```
R7+WH=992a48656c6c6f20576f726c64\r
```

```
34\r
```

```
OK\r
```

There is a variant of this command which will also transmit the current GPS position, along with the supplied data:

```
R7+WHP=...
```

The maximum payload size is 268 bytes (or 250 bytes if using R7+WHP).

## Checking message status

This allows the application to query the status of a submitted message, to see whether it's been sent successfully.

```
R7+STAT=34\r
```

```
1\r
```

```
OK\r
```

Possible returned values are 0 (message not sent yet), or 1 (message sent successfully).



## Receive data

Messages can be sent to the HawkEye 6300 to be consumed by your remote application.

To check for messages waiting, and collect the oldest message received, issue this command:

```
R7+RH\r
```

```
12,486579204A756465\r
```

```
OK\r
```

The response includes a message ID, followed by the hex payload of the message. If no messages are available, it will simply return OK.

You should call this method periodically, and continuously until all messages have been received. To

acknowledge receipt of a message, and to delete it from the HawkEye 6300, issue:

```
R7+D=12\r
```

```
OK\r
```

Where 12 is the message ID to delete. This will make a subsequent call to R7+RH return the next available message.