Getting Started Guide

1. Apply power to the DroneCell

Apply power(5VDC to 16VDC) from a 1 amp or higher power source. Be sure to connect the GNDs together. Note that there are two Power pins and two GND pins. You need to connect to at least one of the Power and at least one of the GND pins.

2. Connect UART pins

Hook up your microcontroller or serial interface's Rx and Tx pins to the appropriate pins on the DroneCell. Your microcontroller's Rx pin connects to the µRX , your microcontrollers's Tx pin connects to the µTX. Make sure your UART is running at 3.3V or 5V TTL.

3. Configure UART

Configure your UART output/input on your microcontroller or serial inteface to the following: Baud rate is 115200, no parity, no flow control.

4. Connect the Antenna.

Plug in and connect your antenna to the antenna connector.

5. Boot up DroneCell

Push the PwrKey button down for three seconds or, alternatively, pull the PWRKey pin LOW for three seconds. Then let go or bring it high. An LED should start flashing now, signaling network connection. Fast blinking means searching for network, slower blinking signifies a connection to the cellular network.

Immediately when the DroneCell boots up, you should receive these startup messages over UART.

RDY
+CFUN: 1
+CPIN: READY

After a few seconds of initialization, you'll receive this message too

Call Ready

At this point you'll be able to communicate with the module over UART
Note that each command needs to be followed by a carriage return, \r , which is basically your enter key.
A sample command is 'AT', which will return 'OK' in the terminal window.

PIN DESCRIPTION

GND   Ground connection
µTX   UART Rx of the DroneCell, connect to Tx of microcontroller
Pwr   +5VDC - +16VDC power supply
PwrKey   Power ignition signal. Pulse HIGH-LOW-HIGH to turn on/off
NetLight   Network status light, connected to status LED onboard
µRX   UART Tx of the DroneCell, connect to Rx of microcontroller
SIM Switch   SIM card switch for holders with built-in switches
Status   Power status of the DroneCell

DRONECELL DEFAULT SETTINGS:
Baud Rate: 115200
Local Echo Enabled
Short Response Enabled

AT COMMAND SYNTAX
The "AT" or "at" prefix must be set at the beginning of each Command. To terminate a Command line enter <CR> , otherwise known as carriage return or \r.
Commands are followed by a response that includes <CR><LF><response><CR><LF>. Only the responses are presented in the document here, <CR><LF> are omitted intentionally.
Example:
With Local Echo enabled:
       Transmit: AT\r
       Receive: AT\r\r\nOK\r\n

SETTINGS FOR EASY SERIAL TERMINAL COMMUNICATION
Nothing extra needs to be done for easy communication with a serial terminal program. All error codes are easily readable and very understandable. Everything is in pretty much regular English.
Local echo, or the setting that shows what you type, is enabled by default. So is the setting for long response to commands.

SETTINGS FOR EASY MICROCONTROLLER COMMUNICATION
When communicating with the DroneCell using a microcontroller, you usually want very short responses, no local echo, and no startup messages. Also, you may want to lower the baud rate from 115200 if your microcontroller cannot handle that high a serial rate.

Sticking on the ;&W to the end of the command saves the setting into memory.

     Enable short response
        ATE0;&W\r

     Disable Local Echo
        ATV0;&W

     Set Baud Rate to 38400
        AT+IPR=38400;&W

     Disable "CALL READY" Startup Message
        AT+CIURC=0;&W

  Now instead of commands returning OK or ERROR in plain text, as well as repeating all written commands, the DroneCell will not echo what you transmit and the DroneCell will return error codes in single bytes. For example, instead of:

Transmit: AT\r
Receive: \r\nOK\r\n

You'll have:

Transmit: AT\r
Receive: \r\n0\r\n

Here are the differences in responses, from long response to short response.

OK	->	0
CONNECT	->	1
RING	->	2
NO CARRIER	->	3
ERROR	->	4
NO DIALTONE	->	6
BUSY	->	7
NO ANSWER	->	8
CONNECT OK	->	8

SAMPLE AT COMMANDS CODE

PHONE COMMUNICATION
Goal: Call a phone
Dial 123-456-7890

> ATD1234567890;\r

> This command returns OK or ERROR. Returns NO CARRIER when phone hangs up

Goal: Answer a phone call

SEND TEXT MESSAGE
Goal: Send a text

AT+CMGF=1\r

> Returns OK or ERROR

AT+CSCS="GSM"\r

> Returns OK or ERROR

AT+CSCA="+13123149810" \r

> Returns OK or ERROR. This number +13123149810 is the short message center for AT&T/Cingular service. T-Mobile's is +12063130004

AT+CSMP=17,167,0,240\r

> Returns OK or ERROR. These numbers refer to settings for text message sending, keep them this way.

AT+CGMS="1234567890"\r

> Returns > , prompting what message to send. 1234567890 is the phone number that the text message will be sent to.

Hello this is a message <Ctrl+z>

> Type any message, then press <Ctrl+z>. Returns confirmation message and Message ID number

Goal: Read a Text

AT+CMGF=1\r

> Returns OK or ERROR

AT+CMGDA="DEL ALL"

> Delete all texts

AT+CNMI=0,0

> Disable unsolicited error code

AT+CMGR=1

> Read Message #1

AT+CMGL="REC UNREAD"

> Read all received unread messages

GPRS TCP COMMUNICATION

Note: All commands have a \r (carriage return) at the end
Goal: Send and receive data over the internet to/from a remote server
// TODO: talk about hardware setup?
Note: TCP server should be running on the remote computer/server
Note: Settings used are for Cingular/AT&T service

1. Power up the DroneCell
2. … Wait for startup messages and until the network status light shows that DroneCell is connected to cellular network (blinks less frequently). Alternatively you can check cellular network status by using the CREG command
   AT+CREG?\r
   Returns network connection status in the form of a number
   This command returns one of six numbers:
   0: Not registered. Not searching for network
   1: Registered to cellular network
   2: Not registered. Searching for network.
   3: Registration denied
   4: Unknown error
   5: Registered, roaming

3. After the DroneCell is powered up and ready, you'll need to set up the access point settings for your network. Be sure you use the right command for your cellular service.

   AT&T/Cingular:
       AT+CSTT="wap.cingular","wap@cingulargprs.com","cingular1"\r
   T-Mobile:
       AT+CSTT="m2m.t-mobile.com","",""\r
   This command will respond with either OK or ERROR

4. And just one more setting to be set. This setting attaches the IPD header, which is useful when receiving data

   AT+CIPHEAD=1\r
   This command will respond with either OK or ERROR

5. Next you'll have to attach or connect to the actual cellular network. This can take over 30 seconds

> **AT+CIICR\r**
> This command will respond with either OK or ERROR

6. After connecting to the cellular network, you now need to get an IP address assigned from the cellular network.

> **AT+CIFSR\r**
> This command will respond with the IP address

7. At this point, all the settings are set, and the DroneCell is all ready for communication with a server. Note in this example that 66.249.90.104 is the IP address of the server and 1024 is the port that you're communicating over.

> **AT+CIPSTART="TCP","66.249.90.104","1024"\r**

8. Now you should wait for the CONNECT OK response ……
   If you get an error , CONNECT FAIL, make sure that your antenna is connected and your server is running.

9. Once you're connected to the server, you're ready to send and receive some data. To send data , use the AT+CIPSEND command
   > **AT+CIPSEND\r**
   > This command returns > or ERROR.

10. Then wait for > response. Once you receive this response you'll be ready to transmit data. Just type or send any string of data at all. When you're done, send Ctrl+Z to end the transmission and close the socket. Also, some socket programs require that you include a \n at the end of your string, so just include it to be safe.

> **1234567890\n<ctrl+z>**

11. Wait for the SEND OK response. If you receive SEND ERROR, then you did something wrong or the server isn't running properly.

Hyperterminal on Windows, Netcat/Socat on Ubuntu

```
Nc –l –p 1036
Nc localhost 1036
```