IT'S HERE!
ArduPilot Mega 2.0

TAKE A LOOK »
THE APM 2 BOARD

The APM 2 board comes from the factory already soldered and ready to have the firmware of your choice loaded by the Mission Planner. *Don’t plug it into your computer's USB port yet. We’re going to get to that in the next step.* This is just an introduction to the board.

The ArduPilot family

<table>
<thead>
<tr>
<th>Autopilot</th>
<th>ArduPilot (aka “Legacy”)</th>
<th>ArduPilotMega APM 1 – 1280</th>
<th>ArduPilotMega APM 1 – 2560</th>
<th>ArduPilotMega APM 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date of introduction</td>
<td>Q1 2009</td>
<td>Q1 2010</td>
<td>Q1 2011</td>
<td>Q4 2011</td>
</tr>
<tr>
<td>Status</td>
<td>Discontinued</td>
<td>Discontinued</td>
<td>Active</td>
<td>Active</td>
</tr>
<tr>
<td>Processors</td>
<td>atmega 328, attiny</td>
<td>atmega 1280, atmega 328</td>
<td>atmega 2560, atmega 328</td>
<td>atmega 2560, atmega 328</td>
</tr>
<tr>
<td>Onboard sensors</td>
<td>None, External: Thermopiles or optional ArdulMU</td>
<td>3-axis gyro, 3-axis accel, baro, optional mag</td>
<td>3-axis gyro, 3-axis accel, baro, optional mag</td>
<td>6-axis MPU6000 (gyro+accel), baro, mag, GPS</td>
</tr>
<tr>
<td>Datalogging memory</td>
<td>None</td>
<td>2MB</td>
<td>2MB</td>
<td>4MB</td>
</tr>
<tr>
<td>Size</td>
<td>30x50x30mm</td>
<td>40x72x20mm</td>
<td>40x72x20mm</td>
<td>40x65x10mm</td>
</tr>
<tr>
<td>Assembly required</td>
<td>Lots!</td>
<td>Some soldering</td>
<td>Some soldering</td>
<td>None!</td>
</tr>
</tbody>
</table>

Here’s a quick tour of your board:


Your RC receiver channels will plug into the Inputs and your servos or ESCs will plug into the Outputs. Up to eight channels in and out are supported out of the box, although both of these can be expanded if needed by following instructions later in the manual.
The connectors that are soldered on at the factory should be right for most people, but if you want to add more sensors, there are nine ports on the left side (A0 to A8) for that.

Here’s where to connect optional external sensors and controls (not all are supported by all codebases):
=== Alternative ways to power your board ===

The board comes from the factory setup up to be powered by your RC system, with RC input and output power shared. On the bench, you'll probably be powering the board via your USB cable while you set it up and test it. But in your aircraft, you'll need to power it with your onboard power system, which is usually your LiPo battery going through an ESC.

It's also possible to power APM 2 from two separate sources, one powering the RC system on the input side, and the other powering the output side (servos or ESCs). This is determined by a jumper on the JP1 pins (see below). If the jumper is on, which is the factory default, the board is powered from the Output rail. If the jumper is off, the board is powered from the Input rail, but the Output rail will need its own power source. This configuration is used if you want to have two separate power sources in your aircraft, one powering the servos and the other powering the electronics.
The APM2 is designed to properly operate from a clean, well filtered power source. Be aware that the voltage specifications are different depending on which connectors are used. The reason for this is there is a schottky diode (diode D1) that prevents the USB port, when connected, from sending power to the APM2's PWM Output connector and damaging something. Consequently, when the USB is not powered and power is connected via the PWM Output connector there is small voltage drop across this diode in normal operation and a higher input voltage is required to compensate. Therefore, power requirements are as follows: 5.0VDC +/- 0.5V supplied into the PWM input connector, jumper JP-1 removed. 5.37VDC +/- 0.5v supplied into the PWM output connector, jumper JP-1 in place

*Warning: Do not exceed 6.0V DC of power supply input voltage or you will damage your board.*

In some cases it may be a good idea to set the input voltage slightly above the median (but below the maximum) to account for possible voltage drops during momentary high current events.

The APM2 by itself draws relatively little current (200ma range) and a power source capable of providing 300 - 500ma will provide plenty of margin. However, if servos or other power consuming devices are being driven by the same power source you must consider the power requirements for those devices as well and provide plenty of margin to prevent disastrous "brown-outs". For instance, a single digital servo can easily draw 1-5 amps depending on it's size and performance. (Note: ESCs do not consume power from the APM) If you experience spurious resets or other odd behavior it is most likely due to noisy or insufficient power to the APM. As with all logic boards, electrical noise from the motors, servos, or other high current devices on the power source can cause unpredictable behavior. It is recommend that a power
filter such as [http://www.dpcav.com/xcart/Power-Supply-Filter-L-C-Type.html this] or [http://www.readymaderc.com/store/index.php?main_page=product_info&cPath=11_15&products_id=533 this] be used in such conditions.

Too short or long power wires, bad or old connectors, or insufficient current capability of the APM power source can result in a "brown-out" situation resulting in unpredictable operation. This is particularly true in traditional helicopters where the collective servos can draw 3-20 amps in short bursts. The power source must be able to accommodate this without voltage droop or voltage spikes. A quality switching type BEC such as one of [http://www.castlecreations.com/products/ccbec.html these] or one of [http://www.readyheli.com/WRL-HBECM2-Western-Robotics-Hercules-Super-Mini-BEC-G2-_p_36453.html these] can be a solution depending on overall current requirements. Many of these type of regulators are programmable so remember to program them with in the safe operating range of the APM2. Linear voltage regulators are not recommended as they are inefficient and prone to overheating and heat induced failures. APM2 should never be connected directly to a battery of any type.

Power source problems are common and can be insidious and frustrating. Be meticulous. Any autopilot or flight controller is useless and potentially dangerous without good clean power source.
Connecting your RC gear

Turning a RC plane into a UAV is essentially just a matter of putting an autopilot between the RC receiver and the aircraft's servos, so the autopilot can take over control.

The way we do that is to plug female-to-female cables from the RC receiver into APM2's inputs, and plug the servos (and motor controller for electric aircraft) into APM2's outputs.

What you'll need:

- At least a 5-channel RC unit. 7 channels or more is highly recommended
- Female-to-female cables for each channel you'll be using. These are fine.
- A power source. For electric aircraft, this is usually the ESC. For gas/nitro powered planes, your radio will need its own battery/BEC. ArduPilot Mega normally gets its power from the RC system.

Note: You can also power APM 2 from the USB cable. However, this will not power the RC output pins, so if you want to see servos move while testing APM 2 on the bench, you'll need to connect a battery (through an ESC) to one of the output channels. It's fine to have both an ESC and the USB cable connected at the same time.

Inputs
Connect your RC receiver to APM2's Inputs with female-to-female cables in the order shown above. Each channel that you want APM to control should be connected to an Input on the APM board. Typically, the Mode Switch (APM Input 8 for ArduPlane) is connected to any RC receiver channel that you have assigned to a three-position toggle switch on your RC transmitter (for most people, that's channel 5).

(Note: These instructions are for ArduPlane. ArduCopter's mode switch goes to APM input 5, not input 8, see this for more)

The ground (black) side of each connector should be on the outside, closest to the edge of the board, as shown here with either straight or right-angle connectors:
Instructions for PPM-Sum receiver are here.

Note: elevons (combined ailerons/elevator, as used in flying wings) are also supported; just connect them to the first two channels as you would to your RC receiver. You'll tell APM to do elevon mixing in the Mission Planner setup process.

If you're flying a V-tail, this is not currently supported by the APM software. Instead, you can mix your rudder and elevator channels externally with a hardware mixer. Here's a good and very inexpensive ($3) one.

Outputs
Plug your servos and other devices you want APM to control into the matching Output connectors as shown above.

**four channel setup:**
- 1. aileron
- 2. elevator
- 3. throttle
- 4. rudder

**elevon setup:**
- 1. starboard (right) elevon
- 2. port (left) elevon
- 3. throttle

## Installing it in your plane

When you place APM in your aircraft, it is very important that it face the right way. The GPS connector should face forward, and the servo cables face back. The board must also be right side up, with the IMU shield at the top. Like this (note: there's a little arrow on the bottom of the shield that point to the front, too, in case you need a reminder at the field):

When putting APM in your plane, it's important to ensure that it's solidly mounted and doesn't move around in flight. It should also be as close as possible to level when the plane is in its flying orientation. Ideally, you'll want to have it as close to the center of gravity as you can (that's where vibration is the least) and mounted on foam tape onto a solid platform.

### Tips on mounting the GPS module
- It doesn't matter which way the GPS module is oriented, as long as the square antenna side is facing up
- It's best to mount the GPS as far away from radio transmission equipment (like your telemetry or video transmitters) as possible.
- Ideally, the GPS module will have a clear view of the sky. On top of your plane is a good choice.
First-time APM Setup

We recommend that you use the Mission Planner for first-time APM setup. (It's designed for Windows but will run in a Windows VM on a Mac or under Mono under Linux. Or, if you don't want to use either of those, you can use the CLI mode, which will work on any operating system.)

After you've downloaded the firmware, ensure that you've got the right COM port and baud rate (115k) selected and click on Connect in the top right corner. The Mission Planner will now connect via MAVLink.

(If this is the first time you're using APM, the code will format your dataflash memory on startup, which can interfere with the MAVLink connection and report a connection failure. If this is the case, let it sit for a couple minutes and try again.)

Now you can click on the setup button circled in red below:

![Setup Button](image)

This will take you to the configuration screens.
Setup steps

1) If you're using a standard airframe, load a pre-made configuration file

For common airframes such as the Bixler, Skyfun and Skywalker, we have supplied configuration files which are tuned for those aircraft. You can download them here and use the Mission Planner to write them to APM. You'll still need to configure them for your own hardware, however, so continue with the steps below.

2) Calibrate your RC input

![Calibration diagram]

Your transmitter must be on. Ideally, you have already flown your airplane in manual RC mode and adjusted any trim values necessary, so the RC outputs reflect these trim settings; if you haven't already flown your aircraft in RC mode and trimmed it out, you may need to do the RC calibration again later, after you have flown the aircraft (this is easy to do at the field).

Channel assignments are as shown above. When you move the RC sticks, the relevant bars will move. Click on "Calibrate Radio" to set the radio limits. Red bars will appear, and you should move them to their limits for each channel you have connected.

On this screen you can also reverse servos if necessary and set up elevon mode if desired.

Press save when you're done.
3) Set your flight modes

You can choose different flight modes in the air with your RC transmitter's toggle switch, which you should have connected to APM input 8. Full details on the available modes are here. (If you want to get more than three modes from your transmitter, a guide is here) When you move the toggle switch, you will see the green highlight change to a different line. You can use the dropdown menu on each line to assign that mode to a function. Note that Flight Mode 6 cannot be changed from Manual. It's "hardware manual", which means that it's controlled by the failsafe circuit on the APM board to always be able to return you to RC control as a safety measure.
4) Configure your hardware

In this tab, you can tell APM what optional sensors you have connected. Just click the check box for any sensor you're using. (Sonar is not currently supported with APM; it's primarily used for ArduCopter)

For the magnetometer (compass), you have a choice of calibration options once you enable the sensor:

1. You can do nothing, and the code will try to figure out all the offsets and declination by comparing the compass readings with the GPS and IMU readings over time in flight. Pro: No user effort. Con: It takes a few minutes of flying to get right, so the compass is inaccurate at first launch.

2. Manual calibration in the Mission Planner (above). You can enter your Declination as instructed below and then press the "Live Calibration" button and move and rotate your aircraft around for 30 seconds while it records the data and does some math to calibrate the sensor. Pro: It works. Con: It's a little awkward, especially for big aircraft. Also it doesn't reflect the magnetic interference that can occur when the motors are going in flight.

3. Replay a flight log. This is a very cool option, shown above as Log Calibration, where you can just replay a previously recorded flight log (.tlog) and the code will compare the GPS and IMU readings with the compass reading and make the necessary corrections. Pro: Works great. Con: You must have already flown, if you load a .tlog file where you didn't actually fly you'll mess up your calibration and will have to do it again or risk poor flight performance.
To manually enter a declination for your geographic location, you can find the correct value by clicking on the link to open a web browser. Enter your location and it will give you a declination, as circled below:

In this case, I would enter "14.13" into the magnetometer line (the software converts from decimal to degrees/minutes).

5) Now if you switch to the Flight Data tab with MAVLink connected, you will see the artificial horizon moving with the board. Remember to leave the board stationary for 15-20 seconds when you switch into this tab, since the IMU must calibrate first. Once it's done, the HUD will start moving.
Sensor Testing

With MAVlink connected, move the APM board and you should see the artificial horizon move accordingly.

Remember how artificial horizons work: when the aircraft tilts to the right, the horizon tilts to the left. (Just tilt your head and you'll see what I mean). This is normal! Please don't tell us it's reversed ;-)   

In the below example, the aircraft/APM board is tilted a bit to the right, and the full sensor data display is shown ("Tuning" checkbox selected).

Unless you have the GPS module connected, are outside and have GPS lock, the map will not show your current position.

Raw sensor display

You can see the output from all the various sensors by clicking on the "Tuning" checkbox on the bottom right of the display (circled in red in this image). This will bring up the graphical sensor display. If you click on the legend at the top of it (also circled in red), it will bring up a window that allows you to choose which sensors and data outputs to display.
Connection of the Airspeed sensor on APM2
Using the Mission Planner to reverse servo direction and set normal/elevon mode

You can reverse the travel direction of any servo by just checking the "reverse" box for that servo in the RC setup screen as shown here:

To select elevon mode or reverse elevon channels, use the elevon checkboxes at the bottom:
Mini tutorial on setting up Elevon mixing in the Mission Planner

Once manual control is setup correctly with your RC transmitter, then there are eight combinations of the three reverse switches possible. Two of these combinations will produce the correct servo output in manual mode, however, only one will both produce the correct servo output in manual mode and map the transmitter inputs correctly to roll/pitch commands in the autopilot-controlled modes, such as Stabilize. It takes a little trial-and-error to find the right combination, but here's a tutorial:

1. First, set it up in manual mode by setting up elevon mixing on your RC transmitter. It matters which elevon is plugged into which channel! For me on the Skyfun, the left wing ailerons should be plugged into Ch1 and the right wing is Ch2.

2. Now that it's working in manual, connect to your APM board with the Mission Planner. Go through the regular setup process. Note that when calibrating your RC input, don't just move the elevator and aileron sticks to the normal up down, left right positions. Instead, you must move the stick to the CORNERS or the calibration will be wrong and the servos will try to move too far. This is because now that you've switched your RC transmitter into elevon mode, the elevator and aileron inputs are added when the stick is in the corner (full left and full up as an example).

3. Now go to back to the RC setup screen.

4. Put your toggle in stabilize mode and move the plane around to test. You'll probably have to reverse something. On the Skyfun, I had to reverse Elevon Ch1. Just change one thing at a time!

5. If ROLL is functioning properly but PITCH is reversed, or vice versa, try swapping the servo cables around (On both Rx and on APM at the same time).

Details of how this works behind the scenes in MAVLink

To setup reversal and mixing using the EEPROM you first need to disable the use of the dip switches. You do that by setting the option SWITCH_ENABLE to 0. If SWITCH_ENABLE is 1 (the default) then the dip switches will control your mixing and channel reversal options.

Next you need to choose if you want to use elevon mixing or not. If you have a 'normal' plane with separate ailerons and elevator then you should set ELEVON_MIXING to 0 (the default). If you have elevons you should set it to 1.

Non-elevon EEPROM setup

For non-elevon setups (ie. setups with ELEVON_MIXING set to 0), you have 3 parameters that control the servo reversals, one for each channel you can reverse.

The 3 parameters are:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>RC1_REV</td>
<td>aileron reversal</td>
<td>set to -1 for reversal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>defaults to 1 (meaning no reversal)</td>
</tr>
</tbody>
</table>
Elevator reversal set to -1 for reversal, defaults to 1 (meaning no reversal)

Rudder reversal set to -1 for reversal, defaults to 1 (meaning no reversal)

Elevon mixing EEPROM setup

For elevon based setups (where you have set ELEVON_MIXING to 1), you have 3 different EEPROM parameters to setup. They are:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Default Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELEVON_REVERSE</td>
<td>Reverse the sense of the elevon mixing</td>
<td>Set to 0</td>
</tr>
<tr>
<td>ELEVON_CH1_REVERSE</td>
<td>Reverse channel 1 elevon</td>
<td>Set to 0</td>
</tr>
<tr>
<td>ELEVON_CH2_REVERSE</td>
<td>Reverse channel 2 elevon</td>
<td>Set to 0</td>
</tr>
</tbody>
</table>

Please make sure that you do careful ground testing after setting these parameters. Also remember that your RC transmitter must be set up to do elevon mixing, too!

Important notes

- Whenever you change your firmware your EEPROM settings will revert to the defaults if the new firmware has an incompatible EEPROM format. Please use the APM mission planner or your ground control station to save your settings, and carefully check them after any firmware change.

- Make sure you always do ground tests before every flight to ensure your channel mixing and reversals are all correct. Be careful to check that not only are your transmitter controls correct, but that the APM responds correctly to attitude changes in the plane when in stabilise mode.
Using the 3DR Radio for telemetry with APM 2

The 3DRobotics 3DR Radio is the ideal way to setup a telemetry connection between your APM and a ground station. Small, inexpensive and with great range, the 3DR radio uses an open source firmware which allows us to do things that cannot be done with other radios.

Radio Features

- very small size
- light weight (under 4 grams without antenna)
- available in 900MHz or 433MHz variants
- receiver sensitivity to -121 dBm
- transmit power up to 20dBm (100mW)
- transparent serial link
- air data rates up to 250kbps
- MAVLink protocol framing and status reporting
- frequency hopping spread spectrum (FHSS)
- adaptive time division multiplexing (TDM)
- support for LBT and AFA
- configurable duty cycle
- built-in error correcting code (can correct up to 25% data bit errors)
- demonstrated range of several kilometres with a small omni antenna
- can be used with a bi-directional amplifier for even more range
- open source firmware
- AT commands for radio configuration
- RT commands for remote radio configuration
- adaptive flow control when used with APM
- based on HM-TRP radio modules, with Si1000 8051 micro-controller and Si4432 radio module

**Connecting your 3DR Radios**

Important note: You cannot connect via the radios when your APM 2 is also connected via USB (they share the same port). Make sure you disconnect your USB cable from the APM 2 board before attempting a wireless connection.

You will need two 3DR radios, one for your aircraft, and the other for your ground station.

Looking at the above picture you will see that some of the radios have a USB connector, making it easy to connect them to your ground station. It uses a D2XX FTDI driver that you can get here.

The 'aircraft' type have a FTDI six pin header, allowing them to be directly connected to your APM telemetry port. Connect the air modules to APM 2 as shown below:
The radios come pre-configured for a serial rate of 57600, which is the default rate that APM uses for telemetry, but you can change this to any rate you like, either using the AT command set, the APM Mission Planner radio setup interface, or the 3DR Radio Configuration Utility.
Status LEDs

The 3DR Radios have 2 status LEDs, one red and one green. The meaning of the different LED states is as follows:

- green LED blinking - searching for another radio
- green LED solid - link is established with another radio
- red LED flashing - transmitting data
- red LED solid - in firmware update mode

Configuring using the Mission Planner

The latest versions of the APM Mission Planner support configuring your 3DR radios using a simple GUI interface. In the Mission Planner (top right) select the Com port that your "ground" 3DR radio is connected to and 57k as the baud rate. Then press Control-A, and it will open this window. Click on "Load Settings" and it will populate it with data similar to that shown (the remote radio's settings will only show if it is also powered on and connected to APM running current ArduPlane or ArduCopter code).

This is the recommended configuration method for most users.
Serial and air rates 'one byte form'

The SERIAL_SPEED and AIR_SPEED parameters are in the same form that APM uses for the SERIAL3_SPEED EEPROM parameter. It is the rate in kbps, truncated to an integer. So '9' means 9600 baud, '38' means 38400, '115' means 115200 etc.

Choosing the air data rate

The key parameter that controls the range of your radios is the AIR_SPEED. The default is 64 (which is 64kbps) will give you a range of over a kilometre with small omni antennas. The lower you set the AIR_SPEED the longer your range, although lowering the AIR_SPEED also lowers how much data you can send over the link.

The radio firmware can only support 13 possible air data rates, which are 2, 4, 8, 16, 19, 24, 32, 48, 64, 96, 128, 192 and 250. If your application needs a different air rate for some reason then we can potentially add it to the register tables. If you choose an unsupported air rate then the next highest rate from the supported list will be chosen.

What air data rate you choose will depend on the following factors

- what range you need
- what data rate you will be sending
- whether you primarily send in one direction, or both
- whether you have ECC enabled
- whether you have an APM firmware with adaptive flow control

For most telemetry applications you will primarily be sending data mostly in one direction, from the aircraft to the ground station. For most people, the amount of data sent from the ground station to the aircraft is small, just an occasional control packet plus heartbeat packets.

If you are using a joystick to control your aircraft then you will be sending a lot more data from the ground station to the aircraft, and in that case you may find a higher AIR_SPEED is needed, although your range will be reduced.

The ECC parameter makes a big difference to the data rate you can support at a given AIR_SPEED. If you have ECC set to zero, then no error correcting information is sent, and the radio uses a simple 16 bit CRC to detect transmission errors. In that case your radio will be able to support data transfers in one direction of around 90% of the AIR_SPEED.

If you enable ECC (which is highly recommended), then the data rate you can support is halved. The ECC system doubles the size of the data sent by the radios. It is worth it however, as the bit error rate will drop dramatically, and you are likely to get a much more reliable link at longer ranges.

If you have the latest APM firmware (ArduPlane 2.33 or later, or ArduCopter 2.54 or later) then the APM will automatically adapt its telemetry rates to what the radio can handle, by using MAVLink RADIO packets injected into the MAVLink streams by the radios firmware. That allows
you to 'oversubscribe' your link, by setting up a SERIAL_SPEED larger than what the radios can actually handle.

The other factor in choosing the air data rate is the TDM 'sync time'. The two radios need to work out each others frequency hopping pattern. They do this by slowly changing the receive channel while rapidly changing the transmit channel. This process of getting in sync with the other radio takes just a few seconds at high air data rates, but gets slower for low air data rates.

For most amateur UAV applications the default AIR_SPEED of 64 with ECC enabled will be good.

**Error Correction**

As mentioned above, the radios support a 12/24 Golay error correcting code if you set the ECC parameter to 1. This means that for every 12 bits of data the radio will send 24 bits, calculating the bits using Golay code lookup tables. The process is reversed on the receiving end, and allows the radio to correct bit errors of up to 3 bits in every 12 bits send (i.e. 25% bit error rate).

The downside of the ECC option is that it halves your available data bandwidth, but in most cases this is worth it, as you are able to sustain a reliable link over longer ranges. You will also get a lot less 'noise' in the serial stream.

**MAVLink framing**

If you set the MAVLINK option to 1 then the radio will do 'MAVLink framing'. The MAVLink protocol is used by APM for transmitting telemetry data to a ground station. When MAVLink framing is used, the radio will try to align radio packets with MAVLink packet boundaries. This means that if a packet is lost you don't end up with half a MAVLink packet being seen by the receiver. That partial packet would appear as line noise on your ground station's console.

The radio firmware will try to fit multiple MAVLink packets into one radio packet where possible for maximum efficiency. The highest radio packet size is 252 bytes.

The radio firmware supports both the MAVLink 0.9 and the MAVLink 1.0 transmission formats.

**MAVLink reporting**

If you have MAVLINK set to 1, then the radio firmware will also look for MAVLink HEARTBEAT messages coming from the serial connection. If it sees a HEARTBEAT message then it knows that the MAVLink protocol is in use, and it will start injecting MAVLink 'RADIO' status packets into the serial stream.

The RADIO packets contain information about the RSSI (Received Signal Strength Indicator) level at both ends of the link, allowing the ground station or aircraft to take action in case the link quality falls too low.
The RADIO packets also contain information about error rates, and how full the serial transmit buffer is (as a percentage). The latest APM firmware can use this information to automatically adapt the telemetry stream rates to the data rate that the radios can sustain.

**Power levels**

You need to be very careful to configure your radios to stay within the legal power limits of the country you are operating in. The default power level of 20dBm is fine for the US and Australia, as up to 30dBm is allowed by the LIPD class licenses there in the 915-928MHz frequency band for a frequency hopping radio. So as long as your antennas have a gain of less than 10dBi you should be within the ISM rules.

The radio cannot support arbitrary power levels. It can only support the power levels given in the following table:

<table>
<thead>
<tr>
<th>Power (dBm)</th>
<th>Power (milliWatts)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.3</td>
</tr>
<tr>
<td>2</td>
<td>1.6</td>
</tr>
<tr>
<td>5</td>
<td>3.2</td>
</tr>
<tr>
<td>8</td>
<td>6.3</td>
</tr>
<tr>
<td>11</td>
<td>12.5</td>
</tr>
<tr>
<td>14</td>
<td>25</td>
</tr>
<tr>
<td>17</td>
<td>50</td>
</tr>
<tr>
<td>20</td>
<td>100</td>
</tr>
</tbody>
</table>

If you choose an unsupported power level the radio will choose the next highest power level from the above table.

Please carefully check the EIRP (Equivalent isotropically radiated power) power limits for your country, making sure you take into account the antenna gain. The 3DR radio is a 'DIY' radio part and it is entirely your responsibility to ensure any use of it is compliant with local rules.

For example, if your local rules allow for a maximum of 30dBm (1W) EIRP, then if you use a amplifier with a 12dB transmit gain, and an antenna with 3dBi gain, then you will need to set TXPOWER to at most 14.

If you don't know how to calculate it, we've made a tutorial for you here: Understanding dB, Watts and dBm.
Using the AT command set

The 3DR radios support a variant of the Hayes 'AT' modem command set for configuration.

If you connect with a serial console to a 3DR radio at the current serial baud rate, you can tell the radio to enter AT command mode by entering the sequence ‘+++’. To prevent data being seen as the command sequence there is a guard time required, so make sure you type nothing on the serial link for 1 second before and after you enter the sequence.

When you enter AT command mode you will receive a 'OK' prompt from the radio and it will stop displaying data sent from the other radio.

Once in AT command mode, you can give the radio either ‘AT’ commands to control the local radio, or (if successfully connected) you can use 'RT' commands to control the remote radio.

The AT commands available are:

- ATI - show radio version
- ATI2 - show board type
- ATI3 - show board frequency
- ATI4 - show board version
- ATI5 - show all user settable EEPROM parameters
- ATI6 - display TDM timing report
- ATI7 - display RSSI signal report
- ATO - exit AT command mode
- ATS{n}? - display radio parameter number 'n'
- ATS{n}=X - set radio parameter number 'n' to 'X'
- ATZ - reboot the radio
- AT&W - write current parameters to EEPROM
- AT&T - reset all parameters to factory default
- AT&T=RSSI - enable RSSI debug reporting
- AT&T=TDM - enable TDM debug reporting
- AT&T - disable debug reporting

All of these commands, except for ATO, may be used on a connected remote radio by replacing 'AT' with 'RT'.

Perhaps the most useful command is ‘ATI5’ which displays all user settable EEPROM parameters. That will produce a report like this:

```
S0: FORMAT=22
S1: SERIAL_SPEED=57
S2: AIR_SPEED=64
S3: NETID=25
S4: TXPOWER=20
S5: ECC=1
```
The first column is the S register to set if you want to change that parameter. So for example, to set the transmit power to 10dBm, use 'ATS4=10'.

Most parameters only take effect on the next reboot. So the usual pattern is to set the parameters you want, then use 'AT&W' to write the parameters to EEPROM, then reboot using 'ATZ'. The exception is the transmit power, which changes immediately (although it will revert to the old setting on reboot unless you use AT&W).

The meaning of the parameter is as follows:

- **FORMAT** - this is for EEPROM format version. Don't change it
- **SERIAL_SPEED** - this is the serial speed in 'one byte form' (see below)
- **AIR_SPEED** - this is the air data rate in 'one byte form'
- **NETID** - this is the network ID. It must be the same for both your radios
- **TXPOWER** - this is the transmit power in dBm. The maximum is 20dBm
- **ECC** - this enables/disables the golay error correcting code
- **MAVLINK** - this enables/disables MAVLink framing and reporting
- **MIN_FREQ** - minimum frequency in kHz
- **MAX_FREQ** - maximum frequency in kHz
- **NUM_CHANNELS** - number of frequency hopping channels
- **DUTY_CYCLE** - the percentage of time to allow transmit
- **LBT_RSSI** - Listen Before Talk threshold (see docs below)

For two radios to communicate the following must be the same at both ends of the link:

- the radio firmware version
- the AIR_SPEED
- the MIN_FREQ
- the MAX_FREQ
- the NUM_CHANNELS
- the NETID
- the ECC setting
- the LBT_RSSI setting

the other settings may be different at either end of the link, although you will usually set them up the same at both ends.

**Support for different countries/regions**
It is very important that you find out what the local country or region regulations are on frequency, hopping channels and power levels and configure your 3DR Radios correctly for your local rules.

Here is some general information to help you get started.

<table>
<thead>
<tr>
<th>Region</th>
<th>Radio Model</th>
<th>Settings</th>
<th>Standard</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA</td>
<td>3DR 900</td>
<td>MIN_FREQ=902000 MAX_FREQ=928000 NUM_CHANNELS=50</td>
<td>FCC 15.247</td>
</tr>
<tr>
<td>Canada</td>
<td>3DR 900</td>
<td>MIN_FREQ=902000 MAX_FREQ=928000 NUM_CHANNELS=50</td>
<td>RSS-210 Annex 8.1</td>
</tr>
<tr>
<td>Australia</td>
<td>3DR 900</td>
<td>MIN_FREQ=915000 MAX_FREQ=928000 NUM_CHANNELS=20</td>
<td>LIPD-2000 item 52</td>
</tr>
<tr>
<td>Australia</td>
<td>3DR 433</td>
<td>MIN_FREQ=433050 MAX_FREQ=434790 TXPOWER&lt;=14</td>
<td>LIPD-2000 item 17</td>
</tr>
<tr>
<td>Europe (most countries)</td>
<td>3DR 433</td>
<td>MIN_FREQ=434040 MAX_FREQ=434790 TXPOWER&lt;=8 NUM_CHANNELS&gt;=30</td>
<td>ETSI EN300 220 7.2.3</td>
</tr>
<tr>
<td>Europe (most countries)</td>
<td>3DR 433</td>
<td>MIN_FREQ=433050 MAX_FREQ=434790 TXPOWER&lt;=8 DUTY_CYCLE=10</td>
<td>ETSI EN300 220 7.2.3</td>
</tr>
<tr>
<td>United Kingdom</td>
<td>3DR 433</td>
<td>MIN_FREQ=433050 MAX_FREQ=434790 TXPOWER&lt;=8 DUTY_CYCLE=10</td>
<td>IR2030/1/10</td>
</tr>
<tr>
<td>New Zealand</td>
<td>3DR 900</td>
<td>MIN_FREQ=921000 MAX_FREQ=928000</td>
<td>Notice 2007, Schedule 1</td>
</tr>
<tr>
<td>New Zealand</td>
<td>3DR 433</td>
<td>MIN_FREQ=433050 MAX_FREQ=434790</td>
<td>Notice 2007, Schedule 1</td>
</tr>
<tr>
<td>Brazil</td>
<td>3DR 433</td>
<td>MIN_FREQ=433000 MAX_FREQ=435000 TXPOWER&lt;=8</td>
<td>Resolução ANATEL n°506/2008</td>
</tr>
<tr>
<td>Brazil</td>
<td>3DR 900</td>
<td>MIN_FREQ=902000 MAX_FREQ=907500 NUM_CHANNELS=11</td>
<td>Resolução ANATEL n°506/2008</td>
</tr>
<tr>
<td>Brazil</td>
<td>3DR 900</td>
<td>MIN_FREQ=915000 MAX_FREQ=928000 NUM_CHANNELS=26</td>
<td>Resolução ANATEL n°506/2008</td>
</tr>
</tbody>
</table>

We'd be delighted to add more countries to this table! Please post the information on the forums, giving links to the applicable regulations and information on what it all means. Also, please point out any inaccuracies in the existing table.

Note that the above table is for most users without any special license. If you are have an application specific license or have a HAM license then you should know what rules are applicable to you.
Finally, compliance is your responsibility. The 3DR radio is a 'DIY' radio part and you need to ensure what whatever you build meets local regulations. Please check your local rules carefully.

**Available Frequency Ranges**

The following table may be helpful matching your local radio regulations to the two radio models available

<table>
<thead>
<tr>
<th>Radio</th>
<th>Minimum Frequency (MHz)</th>
<th>Maximum Frequency (MHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3DR 433</td>
<td>414.0</td>
<td>454.0</td>
</tr>
<tr>
<td>3DR 900</td>
<td>895.0</td>
<td>935.0</td>
</tr>
</tbody>
</table>

**Duty Cycle setting**

Most users will want to set the DUTY_CYCLE to 100. The DUTY_CYCLE is the maximum percentage of time that the radio will transmit packets.

The reason the duty cycle is included is that some regions of the world allow for higher transmit power or more frequencies if you have a duty cycle below a given threshold. So for example in Europe you can transmit on a wider range of frequencies in the 433 band if your duty cycle is below 10%.

When you set a duty cycle below 100% then your available bandwidth will be reduced, so you will find it will only work well for telemetry at higher baud rates. It is still quite practical to get good telemetry from an APM with a 10% duty cycle, as telemetry traffic is quite 'bursty', so the average transmit time is not generally high anyway.

For example, you can easily receive all telemetry streams at 2Hz with AIR_SPEED set to 128, ECC enabled and a DUTY_CYCLE set to 10.

You can also set a radio to receive only by setting the DUTY_CYCLE to 0. That will work best if you set NUM_CHANNELS to a low number, as otherwise the clock synchronisation will be poor.

**Listen Before Talk (LBT)**

The 3DR Radio can implement 'listen before talk' (LBT) functionality to allow it to comply with a wider range of regional regulatory requirements. LBT is a system where the radio is required to listen for a period of time and see no signal from other radios before it is allowed to transmit. By using a non-zero LBT_RSSI value your radio will become more 'polite', by waiting until everyone else has stopped transmitting before starting to transmit itself.

To enable LBT in your radio you need to set the LBT_RSSI threshold. This is the signal strength that the radio considers to be an indication that the channel is busy. If you set LBT_RSSI to zero then LBT is disabled.
The minimum non-zero setting is 25 which is a few dB above the receive sensitivity of the radio (-121 dBm). To setup LBT_RSSI you need to know what signal level your local radio regulations require for LBT functionality. Each increment in LBT_RSSI above 25 is roughly equal to 0.5 dB above the radios receive sensitivity. So if you set LBT_RSSI to 40 then the radio will consider the channel to be free if the signal strength is less than 7.5 dB above the receiver sensitivity.

Alternatively, you can use this formula to get the received signal strength in dBm:

\[
\text{signal\_dBm} = \left( \frac{\text{RSSI}}{1.9} \right) - 127
\]

This formula is approximate, but quite close. See the Si1000 data sheet for a more precise graph.

You will need to lookup your local regulatory requirements to see what LBT_RSSI setting you should use.

The LBT implementation in the 3DR radio uses a minimum listen time of 5 ms, plus randomised listen time as per the European 9.2.2.2 rules.

Note that in many regions you need to implement LBT in conjunction with AFA (Adaptive Frequency Agility). The 3DR Radio implements AFA as long as you have NUM_CHANNELS set to more than 1.

**Technical Details**

When evaluating if this radio meets your local regulations it may be helpful to know what technology it uses.

The firmware implements frequency hopping spread spectrum (FHSS) with synchronous adaptive time division multiplexing (TDM).

Specifically, the radio divides up the frequency range between MIN_FREQ+delta and MAX_FREQ-delta into NUM_CHANNELS channels. The 'delta' value is a guard range to ensure that we stay well away from the edges of the allowed band. The guard range is set to half a channel width. The channel width is defined as:

\[
\text{channel\_width} = \frac{\text{MAX\_FREQ} - \text{MIN\_FREQ}}{\text{NUM\_CHANNELS}+2}
\]

Additionally, the radio skews the base frequency by up to one channel using a random seed based on NETID. This means that two radios using different NETID numbers use slightly different frequencies.

The radios use GFSK (Gaussian Frequency Shift Keying) for transmission on a particular frequency.
The TDM works by dividing up time into slices, based on multiples of 16 microsecond ticks. The time slicing is designed to give a maximum dwell time on any frequency of 0.4s (this is to meet US regulations). The TDM algorithm then works as follows:

- the EEPROM parameters determine a set of TDM parameters, particularly the transmit window and silence period, both are in 16 microsecond units. You can view the results using AT6.
- the transmit window is scaled to allow for 3 full sized packets to be transmitted
- the silence period is equal to twice the packet latency, for the given data rate
- The two radios synchronise their clocks automatically by adding 13 bits of timestamp information to all packets. The timestamp is in 16 microsecond units.
- Each radio only transmits when it is 'their turn'. So a radio gets one transmit window worth of time, then there is a silence period when neither radio transmits, then the other radio gets its turn. We never have the situation where both radios transmit at the same time
- the transmit channels are organised into a random sequence based on the NETID
- the frequency is changed to the next channel twice for each full TDM round, during the silence periods
- when not transmitting, data that comes in over the serial port is buffered in a 2048 byte buffer
- to prevent the buffer from getting too much data (which increases latency and risks overflow) the radios send information on how full the buffer is to the connected device. The APM code adapts its telemetry rates by small amounts to keep the amount of buffered data reasonable.
- The TDM algorithm is also adaptive, in the sense that when it is the turn of radio A to transmit, it can send a small token to radio B saying "I don't need to send anything right now, you can take the rest of my timeslice". That is how the link auto-balances for asymmetric loads
- during the initial search for another radio, and any time the link is lost, the radios go into a mode where they move the receiving frequency very slowly but move the transmit frequency at the normal rate. This allows the two radios to find each other for initial clock sync. How long this takes depends on the number of channels, the air data rate and the packet loss rate.

In some regions you may need to know the distribution of radiated energy within each channel. That depends on a number of factors, but mostly the frequency deviation used for the GFSK modulation. The following formula will give you an estimate of the frequency deviation:

\[
\text{frequency deviation} = \text{air data rate} \times 1.2 \\
\text{min freq deviation} = 40 \\
\text{max freq deviation} = 159
\]

where frequency_deviation is in kHz and the air_data_rate is in kilo bits per second.

**Using a bi-directional amplifier for very long range**

You can combined a 3DR Radio with a bi-directional amplifier in order to extend the range to very long distances.

We have had a lot of success testing amplifiers made by Shireen. In particular, Shireen were kind enough to donate a 900MHz amplifier which we tested with a pair of 900MHz 3DR radios.
This amplifier gives 12dB transmit gain, and 18dB receive gain, automatically switching between transmit and receive modes when the radio starts and stops transmitting. It can either run on 5V from a UBEC, or can use a built-in switching regulator with a 2S or 3S LiPo.

We tested this amplifier at one end of a 7.6km link between two hills in Canberra, Australia. At one end we had a simple wire antenna, and at the other end we had a cheap eBay 3.5dBi omni antenna, plus the Shireen amplifier.

This was the test rig at the end without the amplifier

With the following setting:

S0: FORMAT=22
S1: SERIAL_SPEED=9
S2: AIR_SPEED=24
S3: NETID=25
S4: TXPOWER=14
S5: ECC=1
S6: MAVLINK=1
S7: OPPRESEND=1
S8: MIN_FREQ=915000
S9: MAX_FREQ=928000
S10: NUM_CHANNELS=50

we found that we got a great link at 7.6 km range. We then progressively lowered the transmit power at each end of the link in order to measure the 'fade margin', which allows us to estimate how far the radios could transmit at full power. We found that the fade margin was about 12dB.
at both ends, which implies that the radios should have been able to sustain a link at approximately 4x the range we tested over.

Note that if you use an amplifier (or high gain antenna) you need to be very careful not to exceed the EIRP level that your local rules allow.

**Monitoring the link quality**

You can use the MAVLink support in the 3DR Radios to monitor the link quality while flying, if your ground station supports it.

The two key message parameters are RADIO.rssi and RADIO.remrssi. The first is the RSSI (signal strength) level that the local radio is receiving at. The remrssi parameter is the RSSI that the remote radio is receiving at.

Here is a typical graph of the RSSI levels for a flight at my local flying field.

The RSSI value scales approximately as 1.9x the dBm signal strength, plus an offset. See the Si1000 data sheet for the exact mapping between RSSI and dBm received signal strength, or use this approximate formula

\[
\text{signal\_dBm} = \left(\frac{\text{RSSI}}{1.9}\right) - 127
\]

The reason the RSSI varies so much during this flight is that the signal is attenuated when the plane is rolled over in a turn as I was using a simple wire antenna in the plane. The RSSI values for this flight were plenty high enough for the link quality to be excellent throughout the flight using the default radio parameters.

**What range can I expect?**

The most common question about a telemetry radio is what range you can get with it. It is also a difficult question to answer, as it depends on so many factors.
We have done a lot of test flights to try to gauge what the practical range of these radios is with small omnidirectional antennas and no amplifiers. Here is a typical result:

In this case the 3DR 900 radios were setup with default parameters, except that the TXPOWER had been set to 2 dBm, which means they were transmitting with just 1.6% of their maximum power. Theoretically, a radios range doubles with each additional 6 dB of transmit power, so the range achieved with this test should be about 1/8 of the range that the radios can achieve at an air rate of 64kbps. That is why the above graph shows the distance in meters times 8. This was
on a tiny SkyFun model, and I wanted to keep the plane where I could see it, which is why I did the test with reduced transmit power rather than just flying it a long distance.

The radios kept a perfect link throughout this flight, so we are confident that these radios will achieve a few kilometers range in practice. In this particular case I was flying with a small 'wire' antenna in the SkyFun and was using a cheap eBay 3.5dBi antenna on the ground station.

Of course, the range would be considerably better if I had dropped the air data rate. I find 64kbps to be a good rate for general use, but I tend to use 24 kbps if I am wanting to test at longer ranges.

The range of these radios has also been confirmed by other users. For example, I was sent a log showing a good link kept over a flight of 4.5km from the base station, using default radio settings for a 3DR 900 radio. That was using a small omni antenna in the plane, and an 8dB patch antenna on the ground station. The signal level in the log suggests it could have gone quite a bit further.

### Upgrading your radio firmware

The firmware for the 3DR radios is open source, and new features are regularly added. You should check for new releases regularly to get the most from your radios.

The easiest way to upgrade is using the APM Mission Planner. Go into the 3DR Radio configuration screen and use the 'Upload Firmware' button.

After you upgrade please carefully check all your settings. A firmware update may change your settings to the default values if the EEPROM format has changed.

We also encourage you to get involved in the development of the firmware. Start by looking at the [firmware source code](#) and contribute some patches!

### Forcing bootloader mode

If you somehow manage to get your radio in a state where you can't upload a new firmware via the Mission Planner then you may need to force the radio into bootloader mode.

The way firmware upload normally works is the planner connects to the radio and sends a AT&UPDATE command to put the radio into bootloader mode ready to receive a new firmware. That only works if the planner can send AT commands to the
radio.

If you can't send AT commands, then you can force bootloader mode by shorting the CTS and GROUND pins on the radio while powering on. The red LED will light up when in bootloader mode.

On the air radios the CTS and GROUND pins are easy to find, as they are marked on the back of the radio (they are two of the FTDI connector pins). On the USB radios it isn't as obvious, so this diagram may help:

After you have the radio in bootloader mode you should be able to upload a firmware.
Using a computer joystick/gamepad instead of RC for manual control

ArduPlane allows you to fly with a computer joystick sending commands via Xbee rather than a regular RC system. To use it, you must have the following:

1. Wireless telemetry, such as the [DIY Drones Xbee telemetry kit](#)
2. A USB joystick or gamepad such as [this one](#) (shown above)
3. A laptop computer at the field

The Mission Planner makes it easy to setup up your joystick/gamepad control. We recommend that you continue to have your RC gear connected, since it's possible to lose the Xbee connection sometimes in flight and you'll need to be able to fall back to regular RC control or switch to "hardware manual" in the case of an autopilot failure. (Plus you may want to retain your RC trim settings, which ArduPlane will do.) However, in most cases you can use the joystick/gamepad for all manual flying, never touching the RC transmitter.

To set it up, do the following: Once the controller is plugged in, just select "Joystick" (circled in red below) and it will open a screen that will allow you to assign functions to different joystick/gamepad buttons.
**Instructions:**

1. Click "Auto Detect" and move the stick or button on your controller that you want to assign to that function.
2. You can reverse any control direction by clicking the "Reverse" box.
3. If you are flying an elevon or flying wing aircraft, the Mission Planner can mix the elevator and aileron channels, the way a RC transmitter does. Just click the "Elevons" check box and it will mix those two channels.
4. When you're done, if you click "Enable", the Mission Planner will start using the joystick/gamepad for manual control. All other functions will work as usual.
5. You can add exponential control just like a RC transmitter by entering a value in the "Expo" field shown above.
6. If you press "Save" your settings will be saved for future use.
7. Any trim settings you've made with your RC receiver are retained (passed through)
8. You can assign autopilot modes (Auto, RTL, Stabilize, etc) to the joystick/gamepad buttons, as shown above.

Adding an On-Screen Display (OSD) board

APM has an inexpensive ($66) companion OSD board called MinimOSD. It reads all the MAVLink data in the APM telemetry stream and overlays it on the video stream if you're using an on-board camera and wireless video transmitter. This is useful if you're flying in First Person View (FPV) mode or don't want to use a laptop at the field to see your telemetry data in the Mission Planner.

Instructions on connecting and using the MinimOSD board via the APM telemetry port are here. Connection photos with APM 1 and APM 2 are below.

**Important note:** You cannot connect use the OSD when your APM 2 is also connected via USB (they share the same port). Make sure you disconnect your USB cable from the APM 2 board before attempting to use the OSD.
MinimOSD with APM 2

**Diagram:**

- **APM telem port (pins sequence from bottom):**
  - IN
  - OUT
  - +5V
  - GND

- **Video Transmitter**
  - GND
  - +12V
  - VOUT

- **FPV Camera**
  - GND
  - +12V
  - VIN

- **Battery**
  - GND
  - +12V

**Note:**
TX is NOT connected if there is also an XBee in parallel in order to use a GSC simultaneously.
How to NOT use the regulator from the second power stage:

**APM telem port**
(see sequence from bottom)

- IN
- OUT
- +5V
- GND

- TX
- RX
- +3V
- BLK

**Note:**
TX is NOT connected if there is also an Xbee in parallel in order to use a GSC simultaneously.

**Video Transmitter**
- GND
- +12V
- WOUT

**FPV Camera**
- GND
- +12V
- VIN

**Battery**
- GND
- +12V

Two solder jumpers have to be closed in order to feed the second stage from APM:

- add a solder blob here
- add a solder blob here
Automatic Takeoff and Landing

ArduPilot Mega can automatically launch and land an aircraft, as part of a mission plan. Here’s how:

Auto takeoff instructions

The basic idea of automatic takeoff is for the APM to set the throttle to maximum and climb until a designated altitude is reached. To cause the plane to execute a takeoff, add a NAV_TAKEOFF command to your mission, probably as the first command. This goal is handled slightly differently depending on what sensors are attached, but the altitude parameter always specifies the altitude that must be attained before the APM will consider its takeoff complete and load the next Must command.

The APM will initially hold the wings level on takeoff, but as soon as a takeoff heading is established, the APM will adjust roll to maintain that heading.

If you do not have a magnetometer:

As soon as the ground speed, as measured by the GPS, exceeds 3 m/s, the takeoff heading will be set to the GPS ground course. This means that, in a crosswind, the APM may turn downwind somewhat during takeoff. Sometimes, the takeoff heading is accidentally set too early and this will cause the APM to turn the plane to an undesired heading during takeoff. To minimize this problem, try not to move the plane after Auto has been engaged except to throw it in the direction of takeoff. Try not to "wind up" by moving the plane backwards before throwing it. As much as possible, try to duplicate the behavior of a catapult launcher.

If you have a magnetometer:

As soon as the ground speed, as measured by the GPS, exceeds 3 m/s, the takeoff heading will be set to the magnetometer's yaw sensor.

If you do not have an airspeed sensor:

The first parameter of the NAV_TAKEOFF command will specify the maximum pitch the APM will target on takeoff. The minimum pitch is automatically set to 5 degrees positive pitch. As the plane increases in speed (as measured by the GPS), its pitch will increase. The exact formula is:

\[
\text{target pitch} = \left( \frac{\text{GPS speed}}{\text{cruise speed}} \right) \times \text{maximum pitch} / 2
\]

If you have an airspeed sensor:
The first parameter of the NAV_TAKEOFF command will specify the minimum pitch the APM will target on takeoff. The APM will adjust pitch to achieve airspeed_cruise (pitch up if airspeed is above cruise, pitch down if airspeed is below cruise), but it will not pitch below the minimum pitch set by NAV_TAKEOFF.

**Auto landing instructions**

To land the plane, simply add a NAV_LAND command to the end of your mission indicating the latitude, longitude and altitude of your desired touchdown point. In most cases, the altitude should be set to 0. During landing, the APM will shut down the throttle and hold the current heading as soon as the plane is within 2 seconds of the touchdown point horizontally, or as soon as the plane is lower than 3 meters above the touchdown point, whichever occurs first.

On approach, the APM will fly normally if you have an airspeed sensor. If you do not have an airspeed sensor, the APM will hold 0 pitch.

**Example**

Here is an example mission around the Sparkfun building that autotakeoffs, goes around the building and then sets up a landing pattern for an autoland. Note that the waypoints kick in once the plane has reached 30m altitude after autotakeoff, and that it lands at 0m altitude (altitude is given relative to home/launch altitude)
Flight modes

Ardupilot has a range of built in flight modes, and will have more as development progresses. Ardupilot can act as a simple flight stabilization system or a sophisticated autopilot. Flight modes are controlled through the radio or through logic, using the events.pde file.

To setup your radio to control Ardupilot's Flight Modes, use the interactive CLI: Setup/Modes.

Note that the modes names may not show up as exactly the below in your Ground Station, due to the limitations of the MAVLink communications protocol. A table of mappings of how these modes display is as follows:

<table>
<thead>
<tr>
<th>APM Mode</th>
<th>GCS Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual</td>
<td>&quot;Manual&quot;</td>
</tr>
<tr>
<td>Stabilize</td>
<td>&quot;Stabilize&quot;</td>
</tr>
<tr>
<td>RTL</td>
<td>&quot;RTL&quot;</td>
</tr>
<tr>
<td>Auto</td>
<td>&quot;Loiter&quot;</td>
</tr>
<tr>
<td>Loiter</td>
<td>&quot;Loiter&quot;</td>
</tr>
<tr>
<td>FBW-A</td>
<td>&quot;FBW-A&quot;</td>
</tr>
<tr>
<td>FBW-B</td>
<td>&quot;FBW-B&quot;</td>
</tr>
<tr>
<td>Circle</td>
<td>(retains previous mode display)</td>
</tr>
<tr>
<td>Takeoff</td>
<td>(retains previous mode display)</td>
</tr>
<tr>
<td>Land</td>
<td>(retains previous mode display)</td>
</tr>
</tbody>
</table>

Modes:

MANUAL

Regular RC control, no stabilization.

STABILIZE

RC control with stabilization; let go of the sticks and it will level.

FLY BY WIRE_A

The autopilot will hold the roll and pitch specified by the control sticks. Throttle is manual. The
plane will not roll past the limits set in the configuration of the autopilot. Great for new pilots learning to fly.

**FLY BY WIRE_B**

Requires airspeed sensor. The autopilot will hold the roll specified by the control sticks. Pitch input from the radio is converted to altitude error, which the autopilot will try and adjust to. Throttle is controlled by autopilot. This is the perfect mode to test your autopilot as your radio in/out is substituted for the navigation controls.

**AUTO**

Aircraft will follow GPS waypoints set by configuration utility. You can also "nudge" the aircraft manually in this mode.

**RTL**

Aircraft will return to launch point and circle there until manual control is regained. You can also "nudge" the aircraft manually in this mode.

**LOITER**

Aircraft will circle in current position. You can also "nudge" the aircraft manually in this mode.

**Advanced modes**

**TAKEOFF**

Auto Takeoff is set by the mission control scripting only. Throttle is manual (it respects the limits of the autopilot settings so if you have 65% throttle as the max, it will not go above 65%). Once the plane is moving faster than a few m/s it will lock onto a heading and hold that heading until the desired altitude is reached.

**LAND**

Auto Land is set by the mission control scripting only. Throttle is controlled by the autopilot. After getting closer than 30 meters, the course will lock to the current heading. Flare, throttle, flaps, gear, and other events can be scripted based on distance to landing point.
Instructions

You can enter waypoints and other commands (see below for the full list). In the dropdown menus on each row, select the command you want. The column heading will change to show you what data that command requires. Lat and Lon can be entered by clicking on the map. Altitude is relative to your launch altitude, so if you set 100m, for example, it will fly 100m above you.

Note that if the "Absolute Alt" box is checked, the altitude used will be altitude above sea level, NOT altitude above your launch position. If that box is unchecked, ALT will be relative altitude, so 100m will be 100m above your "home" altitude, or where you're probably standing.

Default Alt is the default altitude when entering new waypoints. It's also the altitude RTL (return to launch) mode will fly at if you have "Hold Default ALT" checked; if you don't have that checked, your aircraft will try to maintain the altitude it was at when you switched on RTL.

Verify height means that the Mission Planner will use Google Earth topology data to adjust your desired altitude at each waypoint to reflect the height of the ground beneath. So if your waypoint is on a hill, if this option is selected the Mission Planner will increase your ALT setting by the height of the hill. This is a good way to make sure you don't crash into mountains!
Once you are done with your mission, select "Write" and it will be sent to APM and saved in EEPROM. You can confirm that it's as you wanted by selecting "Read".

You can save multiple mission files to your local hard drive by selecting "Save WP File" or read in files with "Load WP File".

**Basic waypoint commands**

A mission file is a little intimidating to the human eye, but is a powerful scripting language for the autopilot. (Again, remember that the GCS will soon take care of all of this for you. You shouldn't have to see it yourself for long!).

You can have as many commands as you want, ranging from pre-programmed ones to ones that you can create. Here are some common ones:

- `{NAV_WAYPOINT n/a, alt, lat, lon}`
- `{NAV_TAKEOFF pitch, target altitude}`
- `{NAV_LAND n/a, alt, lat, lon}`

Note that in the screenshot above, I've planned a mission that starts with an autotakeoff to 20m attitude, the three waypoints at 100m, ending with one that sets up a landing pattern. Finally an autoland finishes the mission at 0m altitude.

**Ground calibration**

Set your transmitter mode switch to "Manual". This is a safe mode in which to start up the system, and it allows the system to wiggle the servos to indicate its status. During ground start there will be a series of servo wiggles (1 set at the beginning, 2 in the middle, and 3 when complete).

When you power on your board at the field, you should leave the plane motionless on the ground as level as possible (in flight position; so if you have a tail-dragger elevate the tail) until the three colored LEDs stop flashing (about 30 seconds). That means that the gyros have been calibrated and the plane is ready to fly (assuming you also already have GPS lock). If your plane is a tail-dragger, you should prop up the tail to level flying attitude during calibration.

After the ground start completes you should wait for GPS lock before flying. If you do not wait for GPS lock the home location will not be set correctly, and the barometric altimeter calibration will be incorrect. It should take less than two minutes to get lock. If you're using the MediaTek module, the blue LED on the module will flash while it's waiting for lock, then turn solid once it has it. Once that happens, the red LED on APM should stop flashing and turn solid. If the blue MediaTek LED turns solid but the red APM LED is still flashing, press the reset button on APM and once it reboots, the red LED should go solid.
BEFORE EVERY FLIGHT: Before you take off, hold your aircraft in your hands and switch to stabilize mode, then pitch and tilt the plane it to confirm that the control surfaces move the correct way to return it to level flight. This will ensure that you haven't accidentally knocked a DIP switch into the wrong position or otherwise reversed a channel. You should do this before every flight, just as you move your control surfaces with your RC transmitter to ensure that nothing's reversed. Failing to do this is the #1 cause of crashes.

First flight

It is highly recommended that you switch into either Stabilize or Fly By Wire mode and observe the behavior of the control surfaces. They should move to return the plane to level when you pitch or roll it.

Unless you have tuned your configuration parameters for your airframe it IS NOT recommended that you take off in any mode other than Manual.

Second flight

For your second flight, change the third mode (position 3 of your RC mode switch) to RTL using the CLI's setup/modes process.

This will test navigation. The aircraft should return to you and orbit overhead at a fixed altitude (which can be set with the Mission Planner).

Once all this has checked out, you can program waypoint missions and test then in Auto mode.

Notes

If you have ENABLE_AIR_START 1 This mode allows for quick autopilot recovery in the air in the case of a reboot or power glitch. The autopilot will restart using values saved in the EEPROM memory. However, this method also allows for undesirable behavior on the ground if you do not have your radio properly configured and turned on.

When gps lock is achieved the autopilot will observe its speed. If it sees that its speed is near zero it determines that it is on the ground and performs a ground start. In this case you are ready to fly as soon as ground start completes as the autopilot already has gps lock.
Tuning For Your Airframe

With the default PID settings, APM will fly most airframes okay right out of the box. But to fly really well, with tight navigation and reliable performance in wind, you'll want to tune your autopilot.

The easiest way to do this is via the PID settings screen in the Mission Planner:

You can also load and save configuration files from this screen. We've provided some configuration files for common aircraft [here](#), but if the airframe you're using is not listed there, here's a brief guide to the recommended tuning process:

Here's our recommended tuning process:

1. Fly in manual mode just to make sure everything is hooked up right
2. On the ground check Stabilization mode. If you bank the plane you should see the ailerons (or rudder for 3 channel) respond to return the wings level. If you pitch the plane you should see the elevator respond to correct the pitch. If you are using 4 channel control the rudder should respond in the same direction as the ailerons. Change REVERSE_ROLL, REVERSE_PITCH and REVERSE_RUDDER to get these correct..
3. Fly in Stabilization mode and verify that the autopilot will indeed return the aircraft to level flight from a banked or pitched condition. If the aircraft oscillates in pitch or roll severely, reduce the P gains (see below).
4. Fly-By-Wire mode (A or B), checking both pitch and roll. The plane should hold the attitude you tell it. The transmitter stick position directly sets the bank and pitch angles.
Start by flying straight and level, then push the stick all the way to one side. The plane should bank to 45 degrees or so and maintain that bank. Decrease 5-1 ("#define SERVO_ROLL_P .006") if it's unstable; increase it if it's too slow. Repeat with pitch, up and down. If things still aren't quite right, look at your settings for HEAD_MAX, PITCH_MAX and PITCH_MIN. If you move the stick to the extreme right the plane should go into a bank of angle HEAD_MAX (and stay there). Does it do this? If it gets there slowly increase the roll servo P gain. If it oscillates in roll decrease the servo gain. Work on pitch in a similar manner. The goal is for the plane to respond in pitch and roll in a crisp manner to your stick inputs. Work on this without worrying about the other gains until when you flick the stick and hold it to the side the plane will respond crisply and as quickly as in manual flight. You will likely only need the P gain, but some airframes will need I gain to hold the desired bank angle well without steady state error.

5. Now, staying in Fly-By-Wire, again throw the stick all the way over and watch to see if it maintains altitude. If it descends increase PITCH_COMP by .10. You should be able to hold a sustained HEAD_MAX bank turn so you neither gain or lose altitude.

6. Now try RTL mode. The aircraft should come back to you. If it's flying away, increase NAV_ROLL_P by 25%. If it's still not responding enough, increase NAV_ROLL_I. If, on the other hand, the aircraft comes back to you but weaves and snakes, decrease NAV_ROLL_P. If that's not enough, increase XTRACK_GAIN.

7. Finally, it's worth noting that if you change your cruising speed significantly you may need to re-tune your gains. So if you've been testing at 1/3 throttle but want to fly at full throttle, you may find that issues return. It's best to test at about the throttle setting you want to normally fly at.

Here are some general guidelines on modifying the PID gain settings

A bit of theory

(from Doug Weibel)

APM is based on a cascaded PID controller. The navigation control loop determines a desired bank angle based on where we are headed and where we want to go. The navigation gains affect this desired bank angle. For example if we are 10 degrees off course then one roll navigation P gain might produce a desired bank angle of 5 degrees, while doubling that gain would produce a desired bank angle of 10 degrees. The second control loop controls the servo to reduce the error between the desired bank angle and the actual bank angle. When I recommend tuning the servo gains in FLY BY WIRE mode it is because in that mode you are basically inputting the desired bank angle with the stick position and isolating this lower level control loop.

Due to the cascaded nature of the control loops there is a mathematical relationship between the gains and the ultimate effect. However the whole point of using the cascaded controller is to make it intuitive and understandable for the user. Just get the servo gains right first so that the system gives you the desired roll and pitch angles, and then work on the navigation gains for optimal navigation performance. It is much simpler and effective to do it in that order.

Disabling the built-in GPS on APM 2

If you have an APM 2 with the built-in Mediatek GPS and would like to use an external and/or different GPS module, you can do so with some simple modifications:
Just solder a blob over these two pins, to short them out:

**Testing**

1. With no external GPS, connect to APM via the Mission Planner. The HUD should report "No GPS". That's good -- it means it can't detect a GPS module. Now disconnect and remove the power from your board/remove the USB cable
2. Connect your external GPS to the GPS port
3. Connect the USB cable and connect via the Mission Planner again
4. Once it connects, it should report "No Fix" (if you're indoors or don't have a fix yet) or "3D Fix" (if you've got GPS fix). Either are great -- it means your new GPS was detected!
Configure 6 flight modes for Futaba T7CP

I bought a cheap RC Futaba T7CP on ebay, but I spent 3 days in order to understand the right configurations of switches to enable 6 flight modes. Here, my solution. Hope useful for the community.
(I used some hints from other blogs)

1. I used SWE and SWA. SWE is a 3-position switch. while SWA is on the left and it is a 2-position switch

2. You have to set End Point, SubTrim and RT in MIX. First, set the ENDPOINT for CH5 to be 50%(UP) and 52%(DOWN). You can tune the values until the pwm value(see point 5.) are right.

3. Then, set the SUBTRIM for the GEAR (CH5) to be 100%

4. Configure a P-MIX ( I used P-MIX1) to be without offset, and with the following rates: 24% and 60% (You can switch between the rates by using the 2-position switch)

5. Exit (END button) and check the pwm values in CLI->test->pwm.

6. Configure you radio in CLI->setup->modes
NOTE: Maybe, in order to not mess the radio setup, you have first to setup radio channel: CLI->setup->radio

I attach some pictures for sake of clarity.

Regards to everyone

1. First set the endpoint for Rudder (CH5)

2. Then adjust the GEAR of CH5
4. Finally, set the mixing program. I choose P-MIX1
5. Set the Master and Slave Channels (I used for Master CH7 and for slave CH5). Set the 2 rates: you can switch between one and the other by toggle the A switch.