

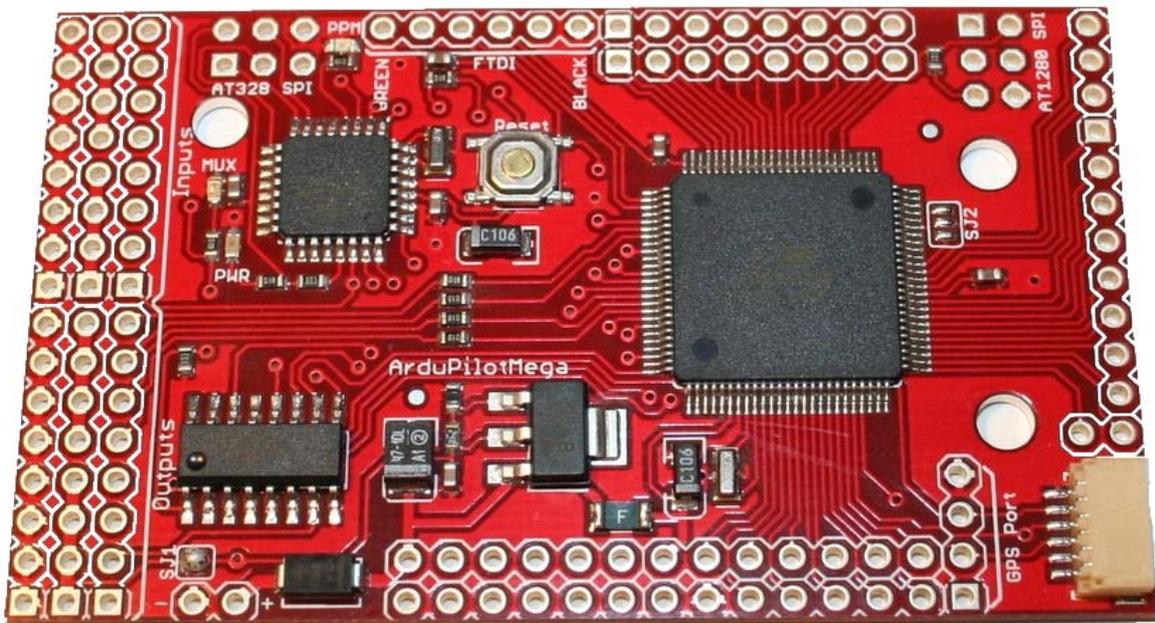
Getting the most out of your APM1280 * by Andrew Tridgell

Do you have an APM1280 board? Are you worried that you won't be able to use it with the latest bleeding edge APM code? Read on for some tips on how to get some life from that old board.

The current generation APM board is based on the APM2560, which has 256k of flash instead of the 128k of flash of the APM1280. As has been discussed before, the rate of code development is such that we are rapidly getting to the stage that if you build with all the latest features enabled you won't fit all the code onto an older APM1280 board. Don't despair though, you can keep using a APM1280 for quite some time!

The first thing to know is that the dev team is working hard on keeping the code size small. In fact, over the last few weeks through some optimisation and code refactoring the total code size of the latest development APM code has shrunk, so it now fits on a APM1280 again with all features enabled.

Development will continue though, and new features are being added at a rapid rate, so it won't be long before the default image grows beyond the 128k boundary again. When that happens you have a few choices for your 1280 board.



Optimise your build for your needs

The key to staying on the bleeding edge with your APM1280 is to optimise the build for your needs. The default options of the APM build are designed to work for as many people as possible out of the box. This means that lots of code is enabled that you won't need on your plane/copter. If you open up the arduino environment and build the image yourself then its easy to disable a few features that you don't need and free up plenty of flash for future development.

To customise your image, just edit your APM_Config.h and add a few lines. Here are some space saving examples:

```
#define GPS_PROTOCOL GPS_PROTOCOL_MTK16
```

Choosing a specific GPS module instead of the default GPS_PROTOCOL_AUTO will save you about 4k of flash. You probably only have one type of GPS anyway!

```
#define LOGGING_ENABLED DISABLED
```

Disabling on-board logging saves more than 10k of flash. Personally I prefer to log my flights to a ground station over MAVLink, so if you are like me and don't need the on-board logging then you can save plenty of flash space.

```
#define CLI_ENABLED DISABLED
```

Disabling the command line interface will save you over 17k of flash. Up till recently it wasn't possible to setup an APM properly without having the command line interface, as you needed it to setup your flight modes and your radio ranges. Thanks to some great work on the APM Mission Planner by Michael Osborne you can now do the complete setup from the planner. That is still a very new feature, so you may find some rough edges, but you can be sure that the dev team are working on making that easier.

There are other examples of how to shrink your build, just take a look in config.h and look for things that are marked as ENABLED by default. If you don't need it then disable it and watch the image shrink.

If you disabled all 3 examples above then you would save over 32k of flash. That's a lot of room for more features, and should keep your APM1280 flying on the bleeding edge for a long time to come.

Using an APM1280 as a HIL simulation board

What if you've decided to get an APM2560 anyway, what can you do with your old APM1280 board?

My favourite use for an old APM1280 board is for HIL simulation. I keep my main APM2560 board in my plane along with my IMU, and I prefer to leave it in the plane between flights, so I'm not constantly having to disassemble things, which can put needless stress on the wires.

My older APM1280 board is kept on my desk, and I use it to fly HIL simulations. The trick to doing this is knowing how to configure your APM1280 to run HIL simulation without an IMU board. I only have one IMU board, and I prefer to leave that in my plane, but the default APM build for HIL will try to access the IMU board, and will crash if its not there.

This is easily solved. The trick is to know that when built for HIL simulation there are only 2 things that the APM code tries to access on the IMU board - the slider switch and the 16MB data log. So if you disable the CLI and the on-board logging then you will find that your APM1280 will fly a HIL simulation perfectly. The defines you will need are:

```
#define LOGGING_ENABLED DISABLED
```

```
#define CLI_ENABLED DISABLED
```

```
#define HIL_MODE HIL_MODE_ATTITUDE
```

The resulting image will be just 66k in size, which means that your APM1280 will be able to fly HIL simulations with the latest bleeding edge code for a long time to come!

To connect your APM1280 for HIL, all you need is a FTDI cable, and of course a flight simulator. I prefer FlightGear, as I can hack on the code, but a lot of people use X-Plane with great success.

So, don't put that APM1280 in your parts bin just yet. There is plenty of life left in it if you tweak things a bit!