

Figure Input & Output Data Flow Diagram

# Autopilot

The autopilot system is by far the most critical system in our design also the most complicated. The required functions of the autopilot are auto takeoff from a flat runway, waypoint navigation and finally landing autonomously at the point of takeoff. There are several different platforms available for accomplishing this task, such as the Openpilot project, the PIXHAWK and the Ardupilot Mega.

The platform chosen is the Ardupilot Mega, primarily because the team managed to find a sponsor willing to provide a complete Ardupilot package for free. It is power by an Atmel AVR ATMega2560 microcontroller which is fast and reliable. There are several advantages of this platform, one such advantage is that the AVR microcontroller is very popular and has a lot of support online. This can be critical in developing the code that would run on this board, as the more help available the easier there is to find answers to questions. Another advantage is that package is completely assembled out-of-the-box; there is no need to get any more electronics and no soldering is required. Furthermore, the autopilot code for the Ardupilot is open-source and well documented thus modifying it would be easier than rewriting the whole stack. Lastly, unlike other autopilot platforms this system incorporates the MPU-6000 IMU rather than a separate accelerometer and gyroscope. The core advantage is that an IMU has a built-in compensator for the accelerometer during banking/pitching. In any system that incorporates an accelerometer, there need to be a method to keep track of the direction of gravity. Primarily because accelerometers report real values, as in if an accelerometer is completely stationary and on a flat surface then a value of 1**G** is register in the zed direction. And it is critical for the calculations of the autopilot to compensate for this, since erroneous acceleration values would make the system fail catastrophically.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Platform | uC | Sensors | Software | Popularity | Cost | Score |
| OpenPilot | 1 | 1 | 1 | 1 | 2 | 6 |
| Ardupilot | 2 | 3 | 3 | 3 | 3 | **14** |
| PIXHAWK | 3 | 2 | 2 | 2 | 1 | 10 |

Table 1 Decision Matrix For Autopilot Platform

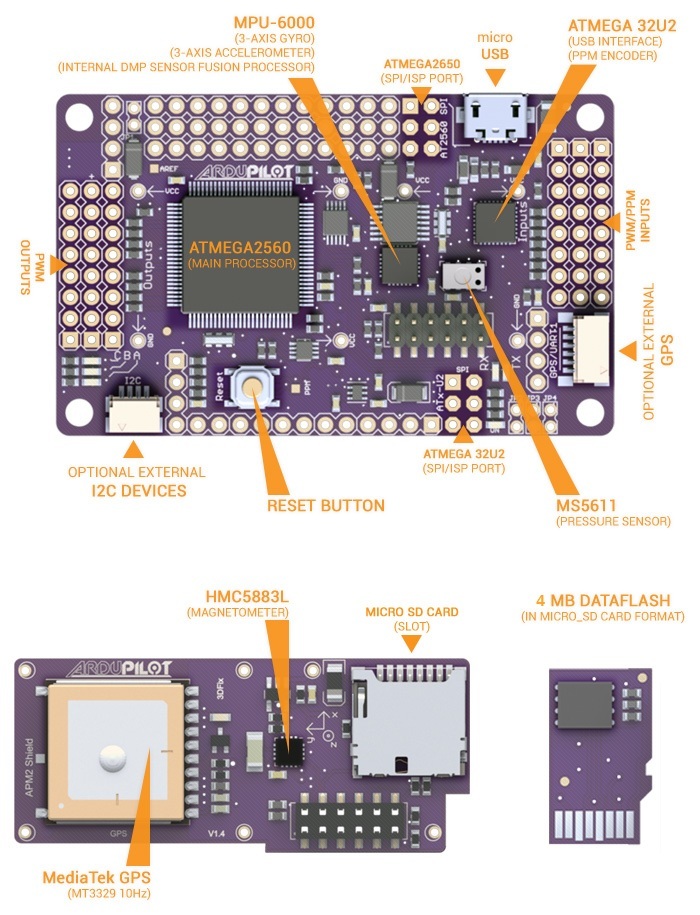


Figure 2 Ardupilot Mega 2.0 Layout [1]

# Autopilot Testing

To test the autopilot regularly is crucial for the overall success of the project; however living in Canada makes it difficult to test outdoors all the time. During the summer and fall seasons it is easy to test the autopilot outdoors in real conditions. The team has chosen an RC field in West Island as our primary testing ground called ‘WIMAC’ (West Island Model Aeronautics Club). The location is far from any major inhabited locations and has a large area for extended flight testing.

However during winter testing outdoors is extremely difficult. Luckily the Ardupilot code allows the developer to recompile the software in HIL (**H**ardware **I**n the **L**oop) mode. Hardware-in-the-loop, allows the developer to modify the code and test it directly on the hardware without the need to place it on a real plane. This is accomplished by simulating the inputs using mathematical formulas to study the outputs of the current algorithms used in the autopilot.

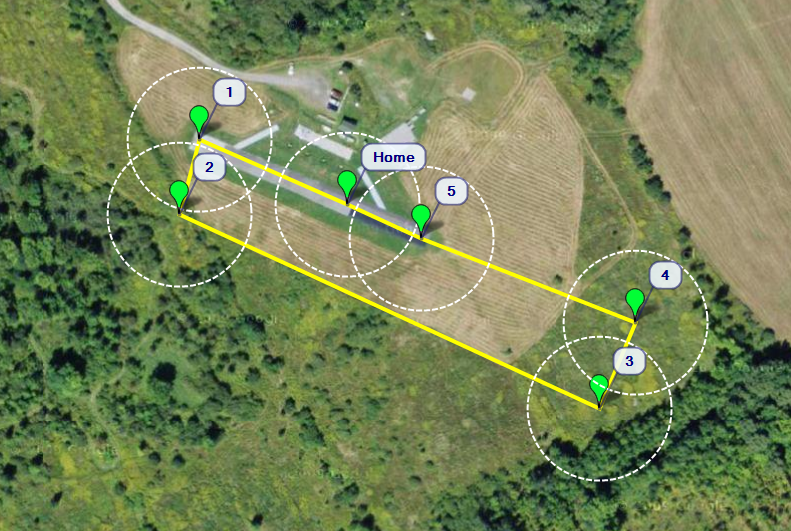


Figure WIMAC Airfield Test Flight Path

# References

[1] “APM2 Ardupilot-mega Official Arduplane Repository”. Internet: <http://code.google.com/p/ardupilot-mega/wiki/APM2board>, Oct 10,2012 [Oct 23,2012]