Contents

[Autonomous System 2](#_Toc352277908)

[Hardware 2](#_Toc352277909)

[Sensors 4](#_Toc352277910)

[IMU (Gyroscope + Accelerometer) [MPU-6000] 4](#_Toc352277911)

[Magnetometer [HMC5883L] 5](#_Toc352277912)

[GPS [MT3329] 6](#_Toc352277913)

[Airspeed Sensor [MPXV7002DP] 7](#_Toc352277914)

[Barometric Pressure Sensor [MS5611] 8](#_Toc352277915)

[Software 9](#_Toc352277916)

[Overview 9](#_Toc352277917)

[Software Architecture 10](#_Toc352277918)

[Attitude and Heading Reference System (AHRS) 11](#_Toc352277919)

[Altitude 13](#_Toc352277920)

[Orientation 13](#_Toc352277921)

[Flight Modes 13](#_Toc352277922)

[References 14](#_Toc352277923)

# Autonomous System

## Hardware

The autopilot system is at the core of the project and thus a lot of time has gone into studying and designing the system. 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.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 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 |

Figure 1: Autopilot Decision Matrix

The platform chosen is the Ardupilot Mega because of a careful study of its individual components and having the most robust design. It is powered 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 reference material is available the easier there is to find solutions to problems. Another advantage is that the 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 needs 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.

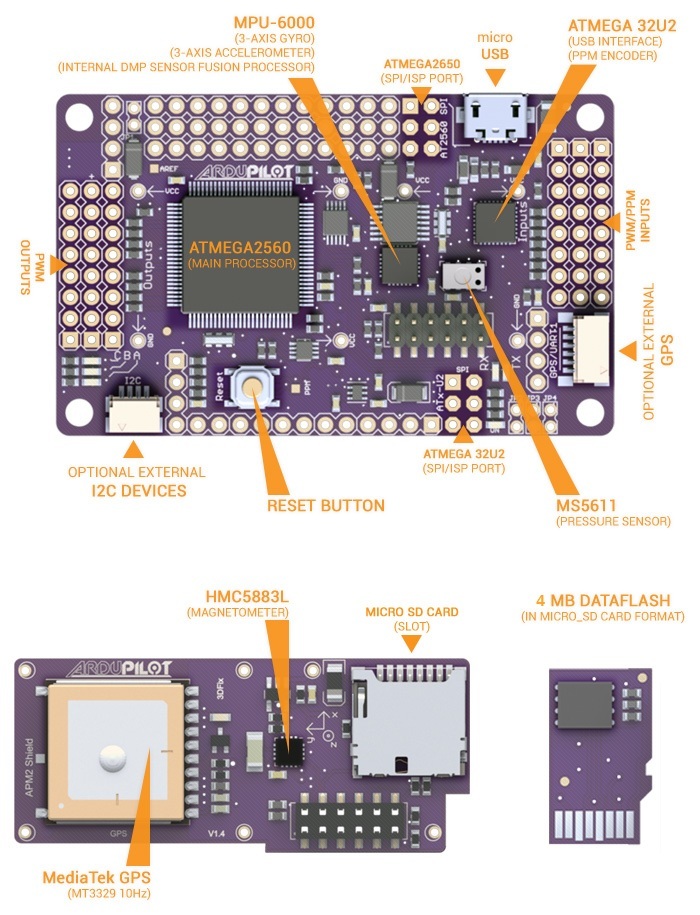


Figure : Ardupilot Platform [1]

## Sensors

### IMU (Gyroscope + Accelerometer) [MPU-6000]

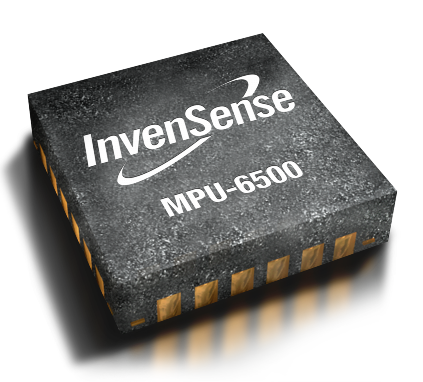


Figure 3: MPU-6000 IMU Module [2]

Table : IMU Specifications [7]

|  |  |  |  |
| --- | --- | --- | --- |
| **Gyro Full Scale Range** | **Gyro Sensitivity** | **Accel. Full Scale Range** | **Accel. Sensitivity** |
| (°/sec) | (LSB/°/sec) | (g) | (LSB/g) |
| ±250 ±500 ±1000 ±2000 | 131 65.5 32.8 16.4 | ±2 ±4 ±8 ±16 | 16384 8192 4096 2048 |

The MPU-6000 includes both a gyroscope and accelerometer in a single package. Also the package includes a small microcontroller that includes several motion sensor algorithms that would allow the implementation of motion detection quickly and efficiently. However, this IMU allows for direct retrieval of raw accelerometer and gyroscope data. This is particularly beneficial for the implementation of more advanced algorithms for motion detection, because airplanes require different algorithms compared to tablets. A gyroscopes primary function is to measure the angular momentum of the airplane during its flight and maneuverings. From the angular momentum it would be possible to integrate the signal and obtain the change in angle. The accelerometer is also critical in navigation because it reports the linear acceleration of all three axes. Again using the values of the accelerometer is essential to obtain the velocity and position of the aircraft.

### Magnetometer [HMC5883L]

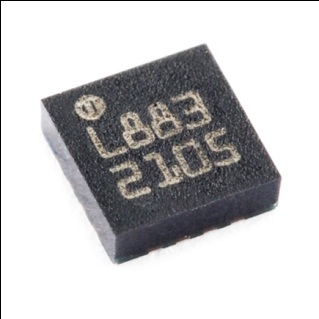


Figure : HMC5883L [3]

The job of a magnetometer is to give the magnitude and direction of any magnetic field. In this case the magnetometer is used detect the magnetic field of the earth, thus the direction of magnetic north. With having magnetic north defined, the magnetometer can be used as a digital compass. Having the heading is critical for designing the auto take-off and auto landing systems. It is due to the need to choose the direction of the runway for the auto landing procedure. Another advantage is the values retrieved during operation are linear, this fact will allow for easy calibration and easy conversion of values for use in the autopilot algorithms.

**Features**

* 1 Degree Accuracy
* I2C Interface
* 160Hz Output Rate
* Low Power (100 uA)

### GPS [MT3329]



Figure : MT3329 GPS Module [4]

The GPS is the most critical sensor for the autonomous flight capabilities of the aircraft, because an accurate state estimation is critical for this system. Without the GPS sensor then there will be no way for the aircraft to find the values of longitude and latitude. Even through the GPS delivers several different data values, such as altitude, velocity and heading. None of the values will be used other than the longitude and latitude because the GPS updates at most at about 10Hz, which is not sufficient for the performance requirement of the aircraft. This GPS also has a built in ARM processor that takes a lot of the processing tasks of the primary controller, thus just delivering for the autopilot a data stream without any overhead saving development time.

**Features**

* 10Hz Update rate
* 3-meter Position Accuracy
* 35 second Cold start acquisition speed

### Airspeed Sensor [MPXV7002DP]

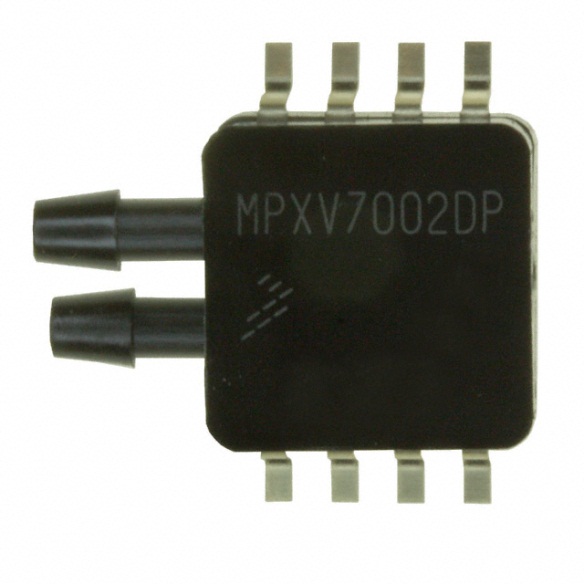


Figure 6: MPXV7000DP Dynamic Pressure Sensor [5]

For finding the airspeed over the wings of the aircraft, there will be a pitot tube attached to a dynamic pressure sensor. The dynamic pressure sensor uses a piezoelectric transducer in the package of the sensor to measure the difference between the static and dynamic pressure, and return the value to the microcontroller. The airspeed sensor is critical for insuring that there is continuous lift over the wing of the aircraft. If there is at any point where the aircraft approaches the airspeed where there is little lift then the aircraft will stall and make recovery difficult. In the autopilot system there are checks and balances that make sure that does not happen.

**Features**

* 2.5% Error over 10C to 60C
* Accuracy of 1.0 kPa
* 1.0 ms Response time

### Barometric Pressure Sensor [MS5611]

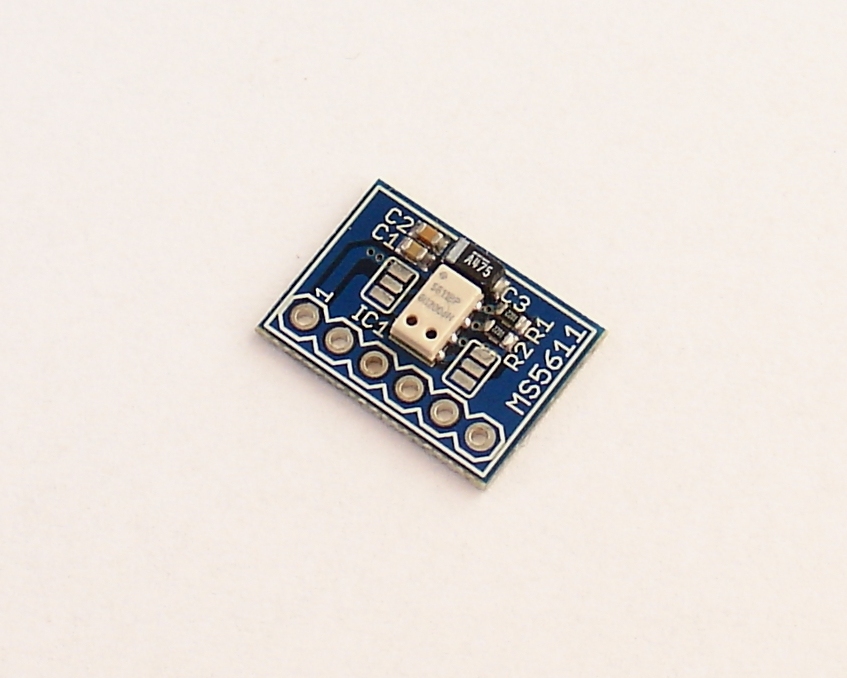


Figure 7: MS5611 Barometric Pressure Sensor [6]

Lastly there is the barometric pressure sensor, which will be used to measure the altitude of the aircraft. This is accomplished by measuring the atmospheric pressure and converting the value using a fixed set algorithm to altitude. This module has a resolution of 10cm, which is quite impressive and is adequate for the systems purpose. Prior to reading the data sheet for this module, there were plans to use a sonar module. There is also an advantage to using this module, with respect to expected noise from the sensor. The characteristic noise of the sensor is about +/- 2.5 mbar which will allow the system to use a less computationally heavy signal filter.

A barometric pressure sensor is built similar to the airspeed sensor, which also measures pressure. A piezoelectric element placed in a plastic enclosure for protection of the elements from impact and debris.

## Software

### Overview

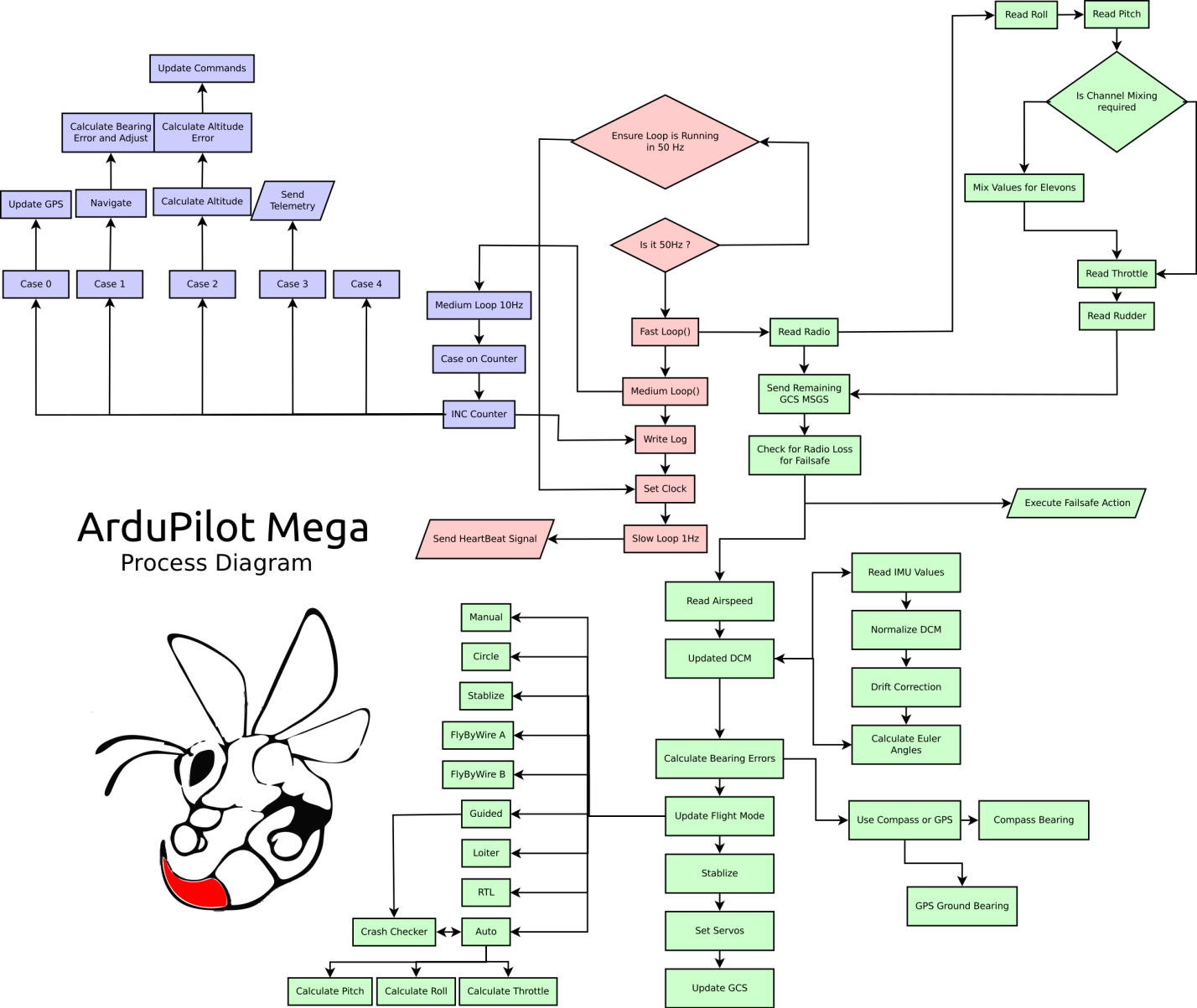


Figure : System Breakdown

There are many stages to any autopilot system these can be split in to two distinct groups estimation and transformation. Estimation is the process by which the system processes sensory input to determine the state of the system. For the aircraft these states are longitude, latitude, altitude, velocity, orientation and heading. The second group of processes is the transformation specific functions which take the estimates derived previously and apply desired transformation to them. From the desired transformation the system can manipulate the actuators to achieve the desired outcome. This all occurs with a feedback from the state estimators to make sure there is minimal error.

### Software Architecture

The Arduplane stack is primarily written in C++, allowing the author to use object-oriented programming to communicate with each component of the system by using instances of each object.

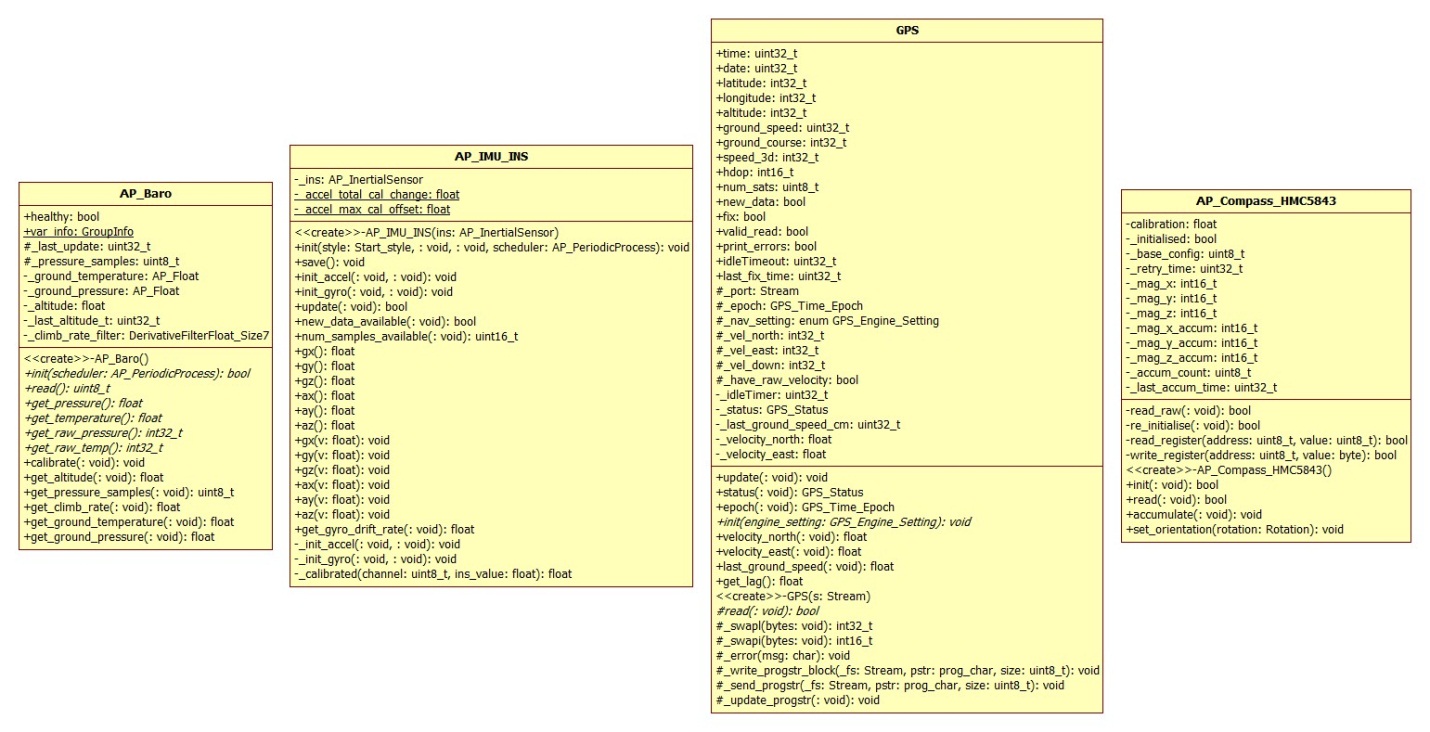


Figure 9: Sensor UML

The software uses a FIFO queue running at 50Hz as a scheduler, timed using a 16-bit timer interrupt. The queue elements are cast as function pointers [**(void\*) funcptr(void)**] and is primarily used to update the readings of each sensor. Each sensor has an **update()** method which is scheduled into the queue and ran to update the readings and accumulate readings to be filtered later on. The actuation of servos, navigation calculations and telemetry is done in another loop running at 3.33Hz.

### Attitude and Heading Reference System (AHRS)

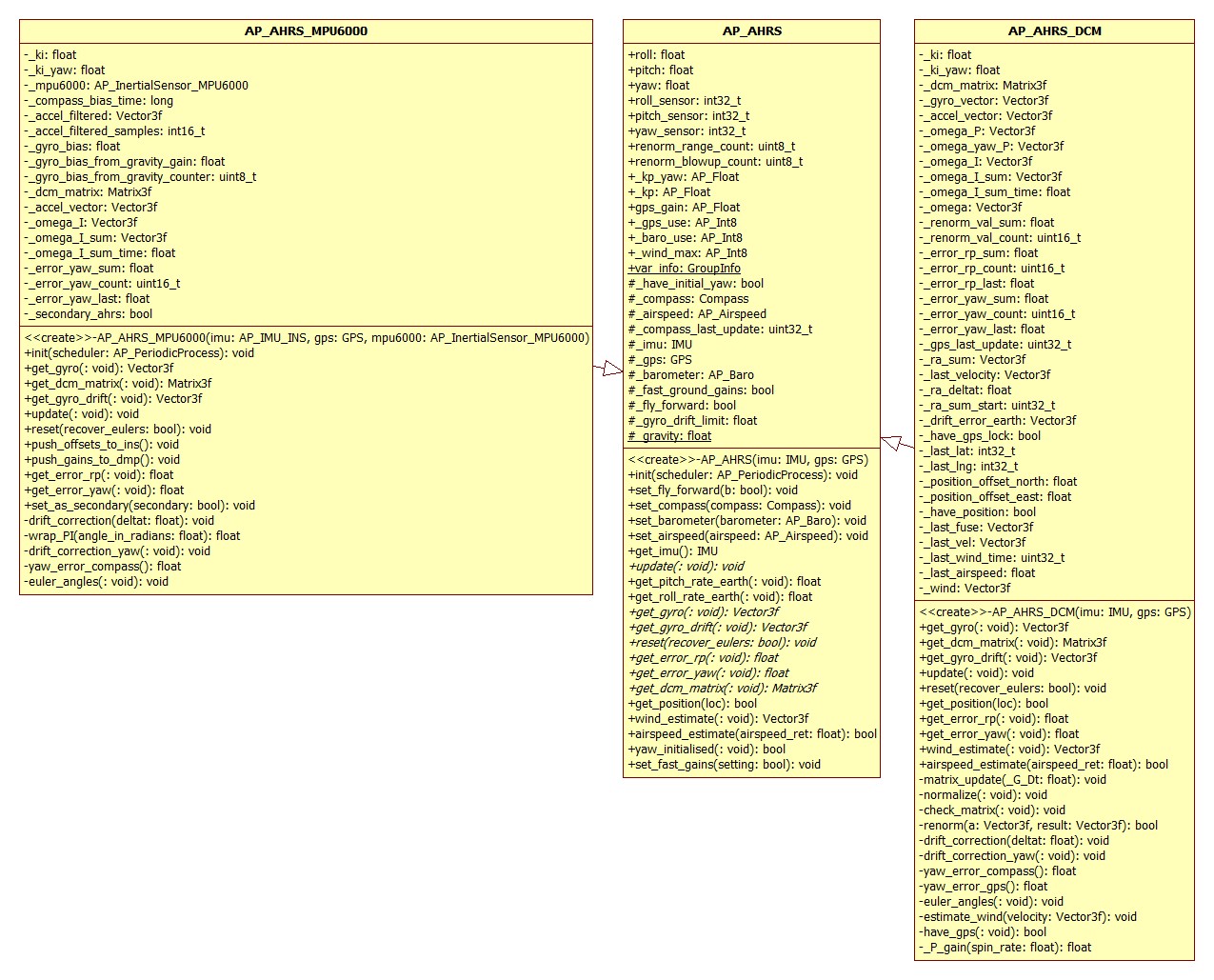


Figure 10: AHRS UML

Most autonomous systems are dynamic in nature and have actuators respond to an input to the system. In the autopilots case the actuators are the servos for the pitch, roll and yaw as well as the throttle. To keep track of the results of these actuations a reference system is needed to find the response of the each input.

For an airplane system three primary references are required for tracking the system as it transforms. These three reference are the planes GPS coordinates, level above ground(ground level above sea level) and the initial orientation of the airplane. With these three it is easy to keep track of the airplane through the entire range of motions.

#### Home Location

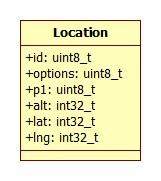


Figure : Location UML

For the home locations longitude and latitude it is simply read from the GPS sensor when there is a fix on the location of the aircraft. The GPS sensor has an error ranging for 3-9 meters and is not accurate but is good enough for our needs.

With respect to the altitude there are two methods to get this information each having their own advantages and disadvantages. Using the GPS to get the altitude is very convenient and do not require any processing, and can be read at the same time as the GPS location. This is done by parsing the NMEA string delivered by the GPS Serial signal. The disadvantage of the using the GPS is the error discussed previously and the time delay in receiving the signal.

The second and preferred method is by using the barometric pressure sensor to calculate the height of the aircraft. The primary advantage for the system is the fast response rate of the measurements (50Hz) and also the high resolution of the sensor allowing for an accurate measurement. The primary disadvantage using the barometric pressure sensor is the requirement for a series of calculations to get the result.

#### Inertial Vector

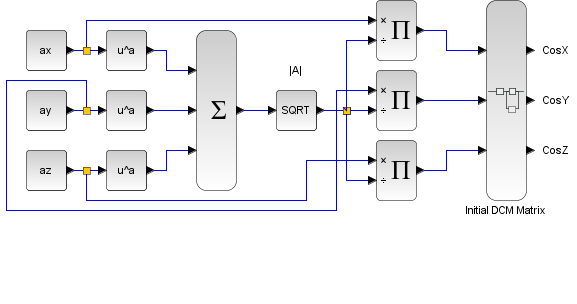


Figure 12: Initial DCM

The initial orientation of the airplane can be calculated by using the above block diagram. Having this initial DCM set is crucial for tracking the orientation of the aircraft through all six degrees of freedom.

### Altitude

Calculating the altitude in the autopilot system is exclusively done through the barometric pressure sensor, the reasoning for this is discussed previously. Solving for the altitude is simply done using the

### Orientation

### Flight Modes

#### Auto Take-off

#### Auto Landing

# References

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

[2] “HMC5883L”. Internet: <http://www.mindkits.co.nz/images/products/sparkfun/10494-01b-435.jpg>, Nov 5,2012 [Nov 5,2012].

[3] “MPU-6000”. Internet: <http://invensense.com/mems/gyro/mpu6000.html>, Nov 5,2012 [Nov 5,2012].

[4] “MT3329”. Internet: <http://store.diydrones.com/MediaTek_MT3329_GPS_10Hz_Adapter_Basic_p/mt3329-02.htm>, Nov 5,2012 [Nov 5,2012].

[5] “MPXV7002DP”. Internet: <http://www.digikey.com/product-detail/en/MPXV7002DP/MPXV7002DP-ND/1168436>, Nov 5,2012 [Nov 5,2012].

[6] “MS5611”. Internet: <http://www.rcgroups.com/forums/attachment.php?attachmentid=4031004>, Nov 5,2012 [Nov 5,2012].

[7] “MEMS Gyro-Accel”. Internet: <http://www.invensense.com/mems/gyro/mpu6050.html>, Nov 5,2012 [Nov 5,2012].