Выбрать платформу для экспериментов с БПЛА сейчас есть из чего. Автопилотов на рынке много, поэтому прежде всего последует небольшой обзор доступных устройств, а затем рассмотрим победителя — Pixhawk (да да, вот так, сразу, никакой интриги).  
  
Критерии выбора:  
  
открытая архитектура (software & hardware);  
современная элементная база (отпадает все по шаблону \*avr\*);  
грамотная, красивая архитектура (отпадают шаблоны \*ardu\*o и \*Rasberi\*);  
механизмы SIL/HIL симуляции;  
наличие открытого ПО наземной станции.  
  
1.Проект Paparazzi  
  
Проект с самой долгой историей (первые AVR версии железа датируются 2003 годом) поэтому начнем с него. В 2010 году погиб основной разработчик проекта Pascal Brisset, но сообщество выжило. К настоящему времени существует более десятка автопилотов на базе Paparazzi.  
  
В исходных кодах много чего накопилось: есть реализация интегрирования кинематических уравнений Эйлера через матрицу направляющий косинусов, углы Эйлера, кватернионы; в float и целых числах. Есть реализация фильтра простого Калмана с 6тью переменными для определения ориентации, а в экспериментальные ветки затесался даже UKF — в общем, есть на что посмотреть. Все это богатство конфигурируется под конкретный аппарат при помощи xml файлов перед компиляцией.  
  
Неполный список устройств на базе проекта есть здесь. Перечислю из него несколько, на мой взгляд, наиболее интересных устройств:  
  
Paparazzi Classix  
  
Концепт 2008года на основе двух LPC2148, чем и интересен. На одном из МК выполняется код автопилота, другой же обеспечивает связь с сервоприводами и приемником ручного управления. Подход уменьшает риск потерять аппарат при отработке автопилота. Канал ручного управления независим, и отказ автопилота не грозит потерей управления.  
  
Paparazzi Lisa/MX V2.1  
  
Современное устройство на безе STM32F4.Содержит полный набор датчиков, за исключением GPS. Цена около 200$.  
  
Paparazzi NavStik  
  
Интересный вариант с модульной структурой. Состоит из очень плотно скомпонованной основной платы с STM32F4 и полным набором датчиков (в том числе GPS и датчика динамического давления) и набора интерфейсных плат.  
  
Набор датчиков уточняется при заказе. Цена основной платы в полной конфигурации 212$.  
  
На интерфейсных платах нет ничего такого что нельзя было бы разместить на основной (разъемы, предохранители). А вот возможность стыковки с Gumstix (с тем что на OMAP3530) очень интересна:  
  
На фотографии этажерка из Gumstix + NavStik + интерфейсная плата.Получается очень серьезное по потенциалу устройство в миниатюрном исполнении.  
  
2.Проект OpenPilot  
  
Проект имеет две аппаратные платформы: CC3D и Revo.  
  
OpenPilot CC3D (STM32F1)  
  
Малогабаритная платка на базе датчика MPU-6000 и МК Cortex-M3. Имеет 6 выходов (ШИМ), 2 UART входа, вход для подключения RC приемника. Обвязки минимум, защиты по интерфейсам нет вообще.  
  
Между автопилотом и железом лежит собственный HAL который, теоретически, можно реализовать через FreeRTOS, но практически написано куча собственно кода под STM32Fx/win/posix. Главный плюс платформы — цена 30$. Доступна, как в официальном магазине, так и на alli, и даже на hk (признание!!!).  
  
OpenPilot Revo (STM32F4)  
  
Устройство на базе STM32F4. На борту датчик давления и модем на 433МГц (для приема помех прямо от контроллера т.к. экран не земечен). Устройство разрабатывали долго, но наконец выпустили. Документация не опубликована, поэтому у китайцев клонов еще нет, у официальных продавцов цена 100-130$ (см. замечание про родной магазин выше).  
  
Математика БИНС реализована через EKF на 13/16 переменных состояния. Фильтр написан на чистом С с использованием float. Код в отдельных моментах оптимизирован (матричные вычисления), в других нет (нормализации кватерниона стоит 4 деления вместо умножения на обратное значение). Непонятно для чего настройки фильтра прописаны прямо в коде (непросто будет под новые датчики настроить).  
  
Ввиду мощного контроллера (168МГц с FPU) для экспериментов устройство подходит значительно больше чем первое.  
  
3.Макетные платы с IMU  
  
Для ряда применений небольшого ARM на борту недостаточно (даже если это М4). Чуть выше уже упоминалась плата Gumstix на OMAP3530 (Cortex-A8) в связке с NavStik. Было бы странно ели бы Gumstix сам не сделал что-то подобное самостоятельно.  
  
Плата расширения Gumstix AeroCore. Другая популярная «макетка» BeagleBone c SOC от TI. Под неё тоже есть подходящая платка:  
  
На фотографии BeagleBone IMU Cape  
  
И, наконец, Intel со своим Edison. Под него платы расширения пока нет, но кто-то сомневается, что будет?  
  
Следует заметить, что перечисленные платформы с трудом проходят по критерию «открытости», вся надежда была на I.MX6, ввиду полноты документации, но его родной SabreLight не подходит по форм фактору, а остальные реализации не имеют должной популярности (сугубо моё мнение). Так что ждем обещанного I.MX6 cо встроенным Cortex-M4 и новой подходящей макеткой.  
  
К отряду макетных плат отнесу так же различные MiniPC на базе Android. Привлекательные, прежде всего, своей ценой. На примере Mk808 за 40$ имеем:  
  
RK3066 2 x Cortex-A9 1.6GHz;  
1Гб ОЗУ;  
8Гб Flash;  
WiFi;  
SD разъем;  
UART на плате;  
  
Есть и более современные устройства на RK3288, но вопросс наличия порта Linux требует дополнительного изучения.  
Подобный miniPC прикрученный через UART к любому описанному здесь автопилоту расширяет его возможности вплоть до VSLAM ( SLAM на основе технического зрения ).  
  
Становится возможным вместо описанных здесь «микро ОС» для МК использовать полноценные ОС  
с надстройками в виде ROS или даже Simulink.  
  
4.Pixhawk Autopilot  
  
Наконец добрались до Pixhawk. Многие видели красивые ролики с квадрокоптерами играющими с мячиком для настольного тенниса, выполняющими акробатические трюки, стаей дружно что-то стоящие. Основной поставщик этого Контента — ETH Zurich (Swiss Federal Institute of Technology Zurich). Pixhawk это их разработка pixhawk.ethz.ch. Благодаря академичности проект имеет красивую архитектуру, как в аппаратной части, так и программной. С аппаратной и начнем. Благодаря открытости приобрести клон можно на Alliexpress, что я и сделал:  
  
Комплектность:  
  
сам автопилот;  
GPS модуль с антенной, батарейкой и магнитометром;  
кнопка со встроенным светодиодом;  
DC/DC преобразователь с датчиком тока;  
переходник USB->UART/I2C;  
разветвитель I2C.  
  
Итого: 185$ (6500р… эх, было время).  
Первый вопрос который возникает: насколько китайский клон «китайский». Что же, вскроем и посмотрим.  
  
Аппаратная часть  
  
Основной блок устройства представляет собой небольшую коробочку со множеством разъемов.  
  
Материал корпуса — полупрозрачный темный поликарбонат  
  
Корпус состоит из двух половин. Половинки корпуса скрепляются при помощи четырех саморезов ввинчивающихся в пенечки на верхней половинке корпуса. Для упрощения сборочно/разборочных операций саморезы были сразу заменены на кусок черной изоленты видимый на первой фотографии. В левом нижнем углу виден небольшой отсек охватывающий датчик давления на плате. До самой платы стенки отсека не доходят. По данным Vooon в родном устройстве в отсеке помещается кусок поролона, но в клоне он почему-то забыт, надо будет исправить самому. Поролон демпфирует колебания давления и защищает датчик от света. ( Спасибо Alex22 за указание на фоточувствительность датчика.)  
  
Основные компоненты смонтированны снизу платы.  
  
Сверху располагаются интерфейсные разъемы, разъем microSD и несколько индикаторных светодиодов.  
К качеству монтажа претензий нет — все ровно, флюс отмыт (кроме батарейки, она, очевидно, паялась вручную после печки). Кстати, у конкурентов батарейки нет (значит нет и часов реального времени, точнее нет пока не пойман GPS).  
Пластиковые разъемы имеют соответствующие отверстия в верхней половинке корпуса. Возле каждого разъема есть подпись. Ответные разъемы входят с некоторым усилием. Для отключения категорически нельзя тянуть за провод (оторвется). Действовать следует так:  
На столе операция не вызывает затруднений, а в поле в -15С можно применить хак:  
  
Имеет смысл сразу запастись запасными кабелями на случай неудачи.  
  
GPS модуль идущий в комплекте совмещает в себе приемник СНС с батарейкой и антенной и магнитометр.  
  
Необходимость выносить магнитометр вместе с GPS антенной возникает так как его желательно располагать подальше от силовых проводников и магнитных материалов. Для примера на DJI это сделано так:  
  
Все входы/выходы выведенные на разъемы имеют защиту от статики, ограничительные резисторы. По цепям питания установлены проходные конденсаторы и защитные диоды.  
  
Для наглядности вот фотография с разметкой части деталей обеспечивающих защиту.  
  
Программная часть  
  
Проект базируется на NuttX RTOS. NuttX представляет собой небольшую posix ОС. Система имеет поддержку ROMFS что позволяет часть настроек производить через конфигурационные файлы при работе системы (сравните со сборкой через конфигурационные xml у paparazzi).  
  
Алгоритмы БИНС основаны на EKF и доступны в виде отдельного проекта InertialNav. Есть варианты реализации под различные наборы датчиков: на 21, 22, 23 и 24 переменных состояния. Код фильтра хорошо документирован и частично получен автоматической генерацией из MATLAB. В проекте имеются исходные MATLAB файлы, архивы с данными — полный набор для опытов.  
  
Работа с устройством «на столе» крайне проста: при подключении к USB в системе появляется новый виртуальный COM порт который можно использовать для связи с автопилотом (в реальном устройстве вместо USB будет подключаемый по UART радиоканал).  
  
Проект совместим с кодом наземной станции управления QgroundControl. Взаимодействи происходит через открытый протокол MAVLink.  
  
Станция управления позволяет настроить автопилот под конкретный летательный аппарат, задать полетное задние и управлять ходом его выполнения, т. е. все что нужно пользователю.  
  
Для пользователя достаточно, но мы пойдем дальше…  
  
Установка среды разработки и загрузка проекта осуществляется в два клика даже из-под windows:  
  
качаем toolchain  
запускаем PX4\_Software\_download.bat — происходит скачивание проекта;  
импортируем проект в Eclipse.  
  
Все, имеем Eclipse c проектом:  
  
Кому приходилось настраивать Eclipse для работы с ARM оценит простоту.  
  
Теперь остается прикрутить ST-LINK и можно начинать отладку… но будет это в следующей статье которая появится только после накопления и осознания соответствующего опыта