

Санкт-Петербургский государственный университет

Кафедра системного программирования

Гальковский Антон Денисович

Интеграция оптических датчиков в квадрокоптер

Курсовая работа

Научный руководитель:
ст. преп. Пименов А. А.

Санкт-Петербург
2019

Оглавление

Введение	3
1. Постановка задачи	4
2. Обзор	5
2.1. Ardupilot	5
2.2. Pixhawk	5
2.3. Датчики VL53L0X	6
2.4. Существующие решения	6
3. Реализация	7
3.1. Подключение датчиков	7
3.2. Избежание столкновений	9
3.3. Интерфейс	10
3.4. Детали реализации	11
3.4.1. Анализ наклона	11
3.4.2. Анализ высоты	12
3.4.3. Мнимое препятствие	12
3.4.4. Буфер замеров расстояний	12
3.4.5. Подключение нескольких датчиков	12
3.5. Отладка	13
Заключение	14
Список литературы	15

Введение

На данный момент использование малых беспилотных летательных аппаратов, а в том числе квадрокоптеров, уже получило достаточную популярность в различных сферах деятельности человека. Тем не менее, остаётся ещё очень большое число областей их применения. В зависимости от решаемой задачи комплектация и стоимость квадрокоптеров может сильно варьироваться. Однако выход из строя во время полёта, приводящий к падению, в любом случае является недопустимым для почти всех моделей. Одна из таких причин – столкновение квадрокоптера с препятствиями. Оно может происходить из-за перераспределения воздушных потоков при подлёте квадрокоптера к стене или из-за неаккуратного расчёта траектории в случае управления человеком с пульта или с помощью предварительно заданного пути движения. Таким образом, избежание столкновений является актуальной проблемой. Одно из классических решений заключается в окружении пропеллеров кольцами, прикрепленными к корпусу квадрокоптера, не позволяющими пропеллерам касаться препятствий. Тем не менее, это решение не предохраняет от перераспределения потоков воздуха, так как эта проблема может быть решена только запретом подлёта квадрокоптера на слишком малое расстояние к стене.

Обобщая вышесказанное, очевидным решением является использование датчиков расстояния, закрепленных на корпусе квадрокоптера. Конечно, такой "оптический барьер" будет малополезным для полётов на открытом пространстве, однако во многих задачах квадрокоптеру необходимо находиться вблизи других объектов. Более того, защита от столкновений делает возможным полёт в небольших замкнутых помещениях.

Системой автопилотирования для реализации данной работы была выбрана система Ardupilot [3], контроллером – Pixhawk [5], как одни из самых популярных. Датчики расстояния были выбраны модели VL53L0X [6] из-за их низкой стоимости.

1. Постановка задачи

Целью данной работы является интеграция (организация подключения и корректировка траектории движения по считанным данным) в систему Ardupilot датчиков расстояния модели VL53L0X. Ввиду предоставления системой Ardupilot большого числа программных инструментов, с помощью которых может быть реализована данная работа, необходимо также провести анализ различных вариантов реализации.

Таким образом, необходимо решить следующие задачи:

- изучение предметной области,
- организация подключения датчиков VL53L0X различными способами,
- организация корректировки траектории квадрокоптера различными способами,
- сравнение полученных вариантов реализации и выбор наиболее оптимального,
- разработка сопутствующего программного обеспечения.

2. Обзор

2.1. Ardupilot

Ardupilot – система автопилотирования с открытым исходным кодом, написанная на C++. Включает в себя программное обеспечение для квадрокоптеров, самолётов, наземных и подводных беспилотных аппаратов.

2.2. Pixhawk

Pixhawk – контроллер, предназначенный для пилотирования квадрокоптеров и других БПЛА и имеющий открытое аппаратное обеспечение. Обладает встроенным акселерометром, гироскопом, компасом и барометром для контроля положения. Имеет много разъемов, реализующих различные интерфейсы (рис. 1).

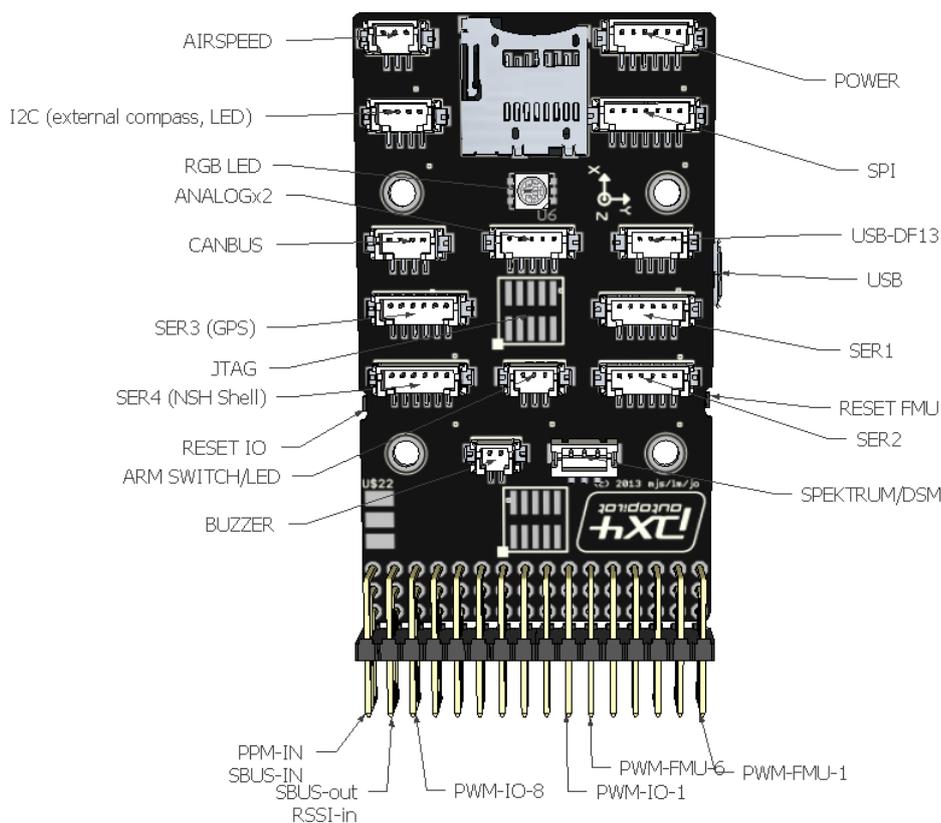


Рис. 1: Схема контроллера Pixhawk

2.3. Датчики VL53L0X

VL53L0X – самый маленький по размеру оптический датчик расстояния (рис. 2), работа которого основана на time-of-flight принципе.

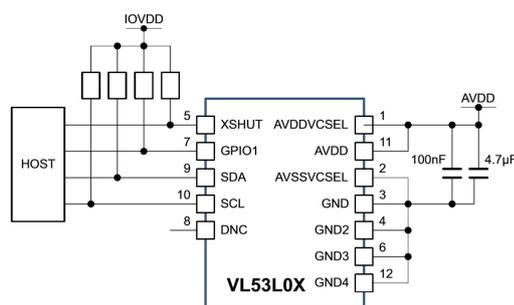


Рис. 2: Схема датчика VL53L0X

2.4. Существующие решения

Для избежания столкновений в системе Ardupilot уже предусмотрено несколько готовых решений [4], использующих датчики расстояния различных моделей. Однако стоимость этих датчиков может в несколько раз превышать стоимость самого квадрокоптера, либо количества датчиков при одновременном подключении может быть не достаточно для стабильного уклонения от препятствий. В частности, датчиков модели VL53L0X нельзя подключить больше одного одновременно, несмотря на то, что угол обзора одного датчика составляет только 25° . Но их небольшая стоимость, даже в случае подключения одновременно шести датчиков, всё же оправдывает их использование (табл. 1). Также алгоритмы избежания столкновений, общие для всех датчиков расстояния, требуют некоторой доработки. К примеру, корректировка положения в горизонтали происходит с помощью только П-регулятора.

Название	Цена
Lightware SF40C	\$999.00 [7]
RPLidarA2/A3	\$599.00 [8]
TeraRanger Tower	950.00€ [9]
VL53L0X	\$2.51 [2]

Таблица 1: Сравнение разных моделей датчиков

3. Реализация

Как было упомянуто выше, решение задачи разбивается на две части: организация подключения датчиков и корректировка траектории движения по считанным данным. Рассмотрим их по отдельности.

3.1. Подключение датчиков

Датчики VL53L0X реализуют интерфейс I^2C , поэтому для считывания значений с одного датчика достаточно просто подключить его в соответствующий разъём контроллера. Но для использования нескольких датчиков одновременно необходимо сначала назначить каждому свой уникальный адрес, притом при следующем использовании каждого датчику потребуется снова назначать свой адрес. Присвоение адреса происходит посредством изменения напряжения на выходе XSHUT одного из датчиков и отправки команды, содержащей новый адрес. Таким образом, помимо общего подключения датчиков к разъёму I^2C должно быть произведено подключение к выходу XSHUT каждого датчика отдельно.

Существует несколько архитектур подключения датчиков:

- с помощью промежуточной платы,
- напрямую к контроллеру.

Первый способ заключается в подключении датчиков к промежуточной плате, занимающейся считыванием с них данных и посылкой контроллеру сообщений с измеренными расстояниями. Второй способ, наоборот, подразумевает общее подключение датчиков к разъёму I^2C самого контроллера и соединение выходов XSHUT датчиков со вспомогательными выходами контроллера.

Промежуточной платой был выбран микроконтроллер Arduino Mega [1] из-за его впечатляющих технических характеристик (в частности, объема памяти и количества различных выводов). Передача данных от

промежуточной платы контроллеру была организована через последовательный порт с помощью протокола MAVLink, включающего сообщение `DISTANCE_SENSOR`, содержащее измеренное датчиком расстояние, его максимальную и минимальную дальность измерения, направление (ориентацию), тип датчика и другие данные.

Итого, были найдены некоторые преимущества каждого из способов. Начнём с первого:

- программа, исполняющаяся на промежуточной плате, обособлена от происходящего в контроллере и может быть реализована произвольным удобным образом (нет необходимости изменять прошивку контроллера);
- нагрузка по опросу датчиков снимается с контроллера;
- в случае возникновения ошибки при работе с датчиками, приводящей, например, к закливанию всей системы, работа контроллера никак не нарушается;
- можно добиться максимальной частоты опроса датчиков, чего нежелательно делать на контроллере, так как в нём для каждой задачи уже предусмотрена некоторая частота её выполнения, чтобы не "отвлекать" контроллер от управления полётом;
- масштабируемость: Pixhawk имеет только 6 выходов, предназначенных для работы в режиме реле, к которым может подключаться выход `XSHUT` датчиков; Arduino Mega же имеет таких выходов 12 штук.

Теперь о втором:

- не происходит увеличения веса всей системы и её физических размеров;
- нет необходимости покупать дополнительную плату, что является существенным фактором, так как важный аспект данной работы составляет вопрос удешевления.

Очевидно, что преимущества одного способа являются недостатками другого и наоборот. Также стоит отметить, что из-за принципа работы датчиков большее время, отведённое на один замер, даёт большую точность измерения, поэтому данные, считанные контроллером (с меньшей частотой, чем на промежуточной плате) будут нуждаться в меньшей фильтрации.

В итоге, преимущества второго способа были признаны более важными, и для финальной реализации был выбран именно он. Ограничение на максимальное число датчиков не сыграло большой роли, так как учет дистанций производился только для 6 направлений (относительно корпуса): вперёд, назад, вправо, влево, вверх и вниз.

3.2. Избежание столкновений

В общем случае хорошим решением для задачи избежания столкновений является использование SLAM алгоритмов. Однако в данной работе подразумевалось решение, не "запоминающее" окружающую обстановку. Данное решение так же может быть алгоритмически устроено разными способами, основанными на одной или нескольких идеях из следующего списка:

- запрет подлёта к препятствию ближе чем на некоторое заданное расстояние,
- ограничение скорости вблизи препятствий,
- ограничение (вблизи препятствий) влияния управляющего воздействия, получаемого с пульта управления, на соответствующее направление,
- использование ПД-регулятора.

Все четыре предложенные идеи в общем случае не являются взаимозаменяемыми, однако при некоторых значениях коэффициентов ПД-регулятора он "покрывает" первые три пункта. Ряд тестовых полётов

квадрокоптера показал, что использование даже отдельно П-регулятора показывает приемлемые результаты при небольших скоростях. Хотя также стоит отметить, что в базовой комплектации (без использования GPS, Optical Flow и других датчиков) в доступный "арсенал" входят только датчики, встроенные в контроллер. Как упоминалось ранее, для контроля положения квадрокоптера в наличии контроллера Pixhawk присутствует гироскоп, акселерометр, компас и барометр. И если барометр вполне справляется со своей задачей, обеспечивая контроль высоты, то за положением в горизонтальной плоскости ни один из перечисленных датчиков следить не может, в результате чего примерную скорость можно рассчитать, например, численно интегрируя данные акселерометра. Однако данный способ является неприемлемым из-за накапливающейся со временем ошибки. Таким образом, более-менее надёжные данные о скорости квадрокоптера можно получить только из датчиков расстояния, но, ввиду того факта, что максимальная дальность измерений датчиков VL53L0X составляет всего 120 сантиметров, гарантию избежания столкновений дать всё же нельзя.

В конечном счёте, были реализованы ПД-регуляторы, использующие измерения датчиков расстояния, для контроля положения в горизонтальной плоскости и для удержания конкретной высоты. В дополнение к этому была добавлена возможность задавать коэффициенты регуляторов, желаемую высоту и канал для включения опции удержания высоты через программу Mission Planner, обладающую удобным пользовательским интерфейсом.

3.3. Интерфейс

С точки зрения интерфейса, результат может быть оформлен в виде:

- добавления в прошивку нового режима полёта,
- изменения одного из существующих режимов полёта,
- добавления в прошивку переменной-флага, отвечающей за корректировку траектории "поверх" существующих режимов полёта.

Главный недостаток первого пункта заключается в том, что в системе Ardupilot уже есть более 20 других режимов полёта и добавление новых только усложнит их использование. Третий способ также не является осмысленным, но уже из-за программных соображений: различные режимы имеют сильно отличающийся интерфейс контроля положения, и для использования описанного флага пришлось бы значительно изменить логику работы всех режимов. Таким образом, был выбран второй вариант. А изменялся, соответственно, режим Stabilize, как один из самых удобных при базовой комплектации квадрокоптера.

Важно заметить, что вызов блока корректировки траектории по данным датчиков расстояния находится непосредственно перед передачей управления блоку контроля наклона (лист. 1). Иными словами, его работа происходит "поверх" остальных блоков.

```
1 angles = get_pilot_desired_lean_angles()
2 /* ... */
3 if (avoidance_enable == true)
4     angles.adjust_using_distance_data()
5 attitude_control.input_angles(angles)
```

Listing 1: Псевдокод вызова корректировки траектории

3.4. Детали реализации

Несмотря на внушительный размер системы Ardupilot (с точки зрения количества строк кода) в ней всё же присутствует большое количество недоработок, некоторые из которых пришлось доделывать в рамках данной работы. Рассмотрим наиболее важные из них.

3.4.1. Анализ наклона

При условии, что квадрокоптер во время полёта может сильно наклоняться, "сырые" измерения с датчиков становятся непригодными для использования в ПД-регуляторе. Но, зная наклон квадрокоптера (измеряемый встроенным гироскопом), можно скорректировать их,

считая, что обозреваемые датчиками предметы имеют только вертикальные и горизонтальные границы.

3.4.2. Анализ высоты

Также на небольшой высоте при большом наклоне несколько горизонтально ориентированных датчиков могут "увидеть" пол, что приведёт к ошибочной корректировке положения. Для решения этой проблемы можно воспользоваться данными о расстоянии до пола, получаемыми с помощью датчика, направленного вниз, и в случае, если горизонтальный датчик "видит" пол, переставать анализировать его измерения.

3.4.3. Мнимое препятствие

Пожалуй, самая грубая из встреченных ошибок заключается в том, что информация о необходимости корректировать положение по каждому направлению обновлялась, только если соответствующий датчик явно обозревал какой-нибудь объект. Это создавало большие проблемы, например, если какой-то объект пересекал область видимости одного из датчиков поперёк.

3.4.4. Буфер замеров расстояний

Как отмечалось ранее, меньшая частота опросов датчиков способствует большей точности измерений, однако избежать необходимость в фильтрации данных таким способом всё же нельзя. Приемлемым вариантом в данном случае стал медианный фильтр, однако подбор размера его буфера оказался непростой задачей: при слишком малом размере фильтрация почти не происходила, а при большом – чрезмерно увеличивалось время отклика.

3.4.5. Подключение нескольких датчиков

Согласно упомянутому выше, датчиков модели VL53L0X нельзя было подключить больше одного напрямую к контроллеру.

3.5. Отладка

В рамках работы была собрана модель для тестирования (рис. 3, табл. 2). К несчастью, во время испытаний происходило много падений из-за сложности в управлении с пульта или ошибок в программе. В конце-концов, повреждения были настолько сильными (в большей степени им подверглись моторы и лопасти), что качественный подбор коэффициентов ПД-регуляторов оказался невозможным. Хотя стоит отметить, что хорошо подобранные коэффициенты не принесли бы почти никакой практической пользы, так как для каждой другой модели их следовало бы подбирать заново. И, как было написано ранее, для этого был добавлен удобный инструмент (см. 3.2).

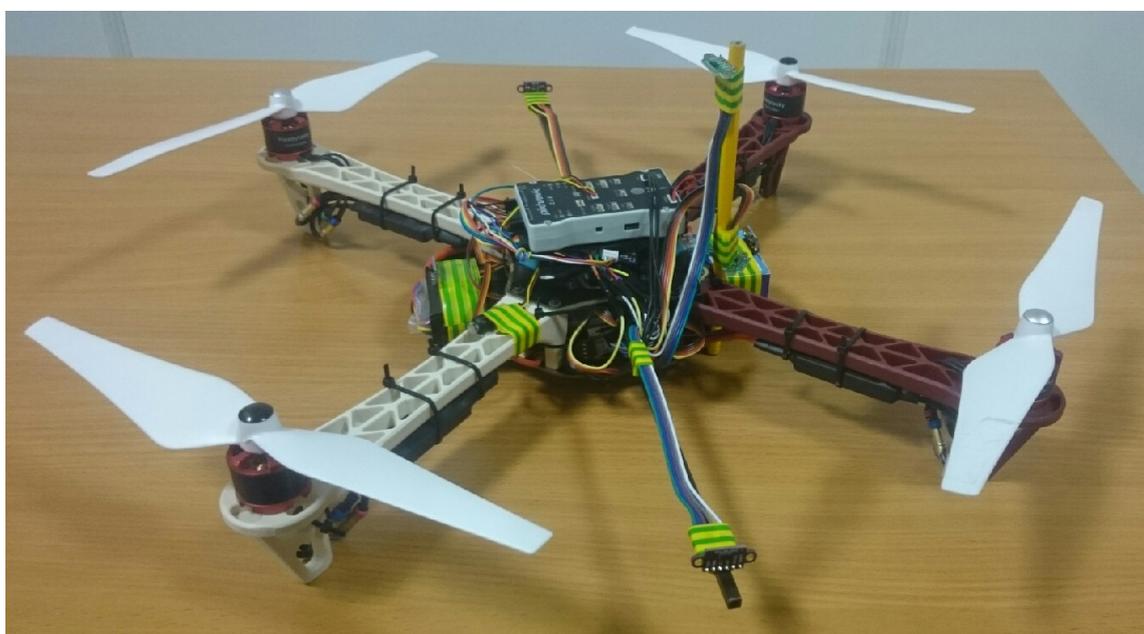


Рис. 3: Внешний вид тестовой модели

Компонент	Количество
Рама F450	1
Контроллер Pixhawk	1
Аккумулятор Lipo 5000mAh 11.1V	1
ESC BLHeli 50A	4
Мотор Readytosky 2212 920KV	4
Пропеллер 9450	4

Таблица 2: Комплектация тестовой модели

Заключение

Итого, в ходе работы были получены следующие результаты:

- изучена предметная область, а также система Ardupilot;
- реализовано подключение датчиков расстояния VL53L0X к контроллеру Pixhawk двумя способами: напрямую и с помощью промежуточной платы Arduino Mega;
- в режим Stabilize добавлен блок корректировки положения, использующий данные с датчиков расстояния;
- устранены некоторые недочёты системы Ardupilot по работе с датчиками расстояния;
- добавлена возможность изменения констант, связанных с работой PID-регулятора, через программу Mission Planner;
- проанализированы возникающие альтернативы, оптимальный вариант внесён в финальную реализацию;
- собрана тестовая модель, на ней испытана алгоритмическая часть избежания столкновений.

Список литературы

- [1] ARDUINO. Arduino Mega 2560 Rev3.— URL: <https://store.arduino.cc/usa/mega-2560-r3> (дата обращения: 15.05.2019).
- [2] AliExpress. VL53L0X Time-of-Flight (ToF) Laser Ranging Sensor Breakout 940nm GY-VL53L0XV2 Laser Distance Module I2C ИС.— URL: <https://clck.ru/EwTNx> (дата обращения: 15.12.2018).
- [3] Ardupilot. Ardupilot.— URL: <http://ardupilot.org/> (дата обращения: 15.12.2018).
- [4] Ardupilot. Object Avoidance.— URL: <http://ardupilot.org/copter/docs/copter-object-avoidance.html> (дата обращения: 15.12.2018).
- [5] Ardupilot. Pixhawk Overview.— URL: <http://ardupilot.org/copter/docs/common-pixhawk-overview.html> (дата обращения: 15.05.2019).
- [6] Ardupilot. ST VL53L0X / VL53L0X Lidar.— URL: <http://ardupilot.org/copter/docs/common-vl53l0x-lidar.html> (дата обращения: 15.12.2018).
- [7] Lightware. SF40/C (100 m).— URL: <https://lightware.co.za/collections/lidar-rangefinders/products/sf40-c-100-m> (дата обращения: 15.12.2018).
- [8] Roboshop. Slamtec RPLIDAR A3 360° Laser Scanner (25 m).— URL: <https://www.robotshop.com/en/slamtec-rplidar-a3-360-laser-scanner-25-m.html> (дата обращения: 15.12.2018).
- [9] TERABEE. TeraRanger Tower.— URL: <https://www.terabee.com/portfolio-item/teraranger-tower-scanner-for-slam-and-collision-avoidance/> (дата обращения: 15.12.2018).