# R3PB AIR TRANSPONDER

Идея проекта возникла, как говорится, не на пустом месте. Так уж вышло, что планеризмом, стал заниматься с 2001 года. По началу «подлётывал» на разных аэродромах, пока в 2007-ом году, по иронии судьбы прям под боком, где я теперь проживаю, обнаружил самый известный в нашей стране аэроклуб «2-ой МАК». С тех пор, можно сказать, осел там же.

Авиационный парк нашего аэроклуба простой, всё как у всех. Остатки былой славы Советского прошлого: Вильги, Бланики, Яки, Янтари которые бережём, своими руками восстанавливаем, и при надобности ремонтируем. Благо коллектив собрался очень дружный и толковый.

Учимся летать, а точнее парить на планёрах. Тут тоже как в любом другом АУЦ, программа обучения – прежде всего буксировка и полёты по кругам до «тошнотиков», несложные пилотажные зоны включая штопорные, и самое главное ради чего всё это – парение. Это когда у тебя немного перья появляются, а инструктор тебя научил восходящие потоки пятой точкой чувствовать при этом самостоятельно можешь лететь. Но увы, летать далеко не отпускают. И это понятно, так как парение — это целая наука! Не рассчитал – не долетел. Сел в поле, хорошо если не на лес. Если что напомню, планер – это всё же как самолёт, но только без мотора. L13 Бланик весит 295 кг без пилота. Разложить об кочки при посадке очень не хочется. Потому то руководитель полётов(РП) внимание уделяет немалое тому, где мы парим, как парим, на какой высоте, в каком потоке, на каком удалении, в каком месте – за стартом или впереди старта… Однако, не всегда есть возможность визуального контроля за бортами. Это бывает по различным причинам. Зона-парение, площадка, у кого маршрут и планёр скрывается с глаз. Есть конечно радиосвязь. Регламент обмена, доклад, и всё такое… Хорошо, но всё же, необходимо больше. Иногда, казалось бы уже «всезнающие» планеристы, всё же увлёкшись потоком забываются, и уходят далеко за старт, да и другие причины так же бывают - азарт во чтобы то ни стало, найти тот самый термик... Как раз из-за подобных ситуаций, родилась идея того, чтобы можно было как-то следить за бортами всех пилотов, особенно курсантов, не обладающих необходимым опытом. Что бы РП мог всегда увидеть, подсказать, дать команду на выполнение. В общем дополнительно обезопасить весь процесс обучения. Так и появилась идея, установить на борт что-то, что может передавать параметры.

Конечно же первое что пришло в голову – это система ADSB и прочие xDSB. Всё это круто, но нереально дорого для нас, да и места нет для такой сложно аппаратуры, ко всему прочему имеющей вес. Другие коммерческие решения, так же оказывались дорогостоящими, либо не соответствующими нашей специфике, да и попросту не удобными.

Ну чтож. Пришлось, применить скромные радиолюбительские навыки.

Первое что было сделано – определено техническое задание от которого и оттолкнулся.

Итак, ТЗ:

1. Первое и самое важное – **никаких сторонних операторов связи**, никакого интернет, GSM, 5G и каких-либо других сетей связи, включая радиолюбительскую APRS. Только своя базовая станция(БС) на земле и транспондер в воздухе, за исключением GPS/ГЛОНАС приёмника – тут уж никак.
2. Определено **количество транспондеров** в системе. Так как авиапарк не большой, то и количество было определено по количеству бортов – **10 транспондеров в системе** (хотя можно и больше).
3. Определён максимальный **радиус действия** системы. Основываясь на опыте полётов наших старожилов, и учитывая требования безопасности полётов, за основу было взят круг радиусом **35 км** (думаю можно больше в раза два). Это расстояние, в которое попадают всё значимые населённые пункты на карте, дальше которых на парение мы не улетаем. Ну и примерно 100 километровый треугольный маршрут для зачёта. Если учесть, что речь идёт о обучении то новички редко за 8 км от старта уходят, то более чем достаточно.
4. Транспондер летательного аппарата должен в автоматическом режиме **передавать**:
   * **Место**
   * **Высоту**
   * **Скорость**
   * **Курс**
   * **Скороподъёмность**
   * **Номер борта**
5. Транспондер должен **передавать данные** не реже **1 раза в 4 секунды**.
6. Мощность транспондера в пределах 1 Ватта. Питание автономное.
7. **Базовая станция**(БС) должна **принимать данные** со всех транспондеров **одновременно**.
8. **Данные** полученные БС должны быть **записаны в базу данных**.
9. **Данные** полученные с транспондеров должны **отображаться на карте в режиме реального времени**.
10. **Отображать данные** можно на компьютере планшете смартфоне в **WEB браузере**. Повторюсь, НИКАКОГО ИНТЕРНЕТА!!! Всё оффлайн.
11. **Базовая станция автономна**: батарея, плюс антенна на СКП. **Подключение** компьютера к БС **с помощью Ethernet или WIFI**.
12. Решение должно быть максимально бюджетным, при этом габариты транспондеров должны быть такими чтоб ничему и никому не мешали.

Первый пункт многое перечеркнул. Если бы не этот пункт то, готовых решений, на просторах интернета, сходу найти можно с дюжину. Всяческих GPS трекеров любительских. Про коммерческие и вовсе молчу. Но всё работает в сетях операторов связи. Это минус. (*Хотя на будущее включить использование такого вида связи в качестве неосновного аварийного всё же стоит*).

Так же не стал привлекательным вариант с использованием APRS системы, а именно с развёртыванием своей сети не включенной в общую сеть. Слишком громоздко. Да и требования пунктов 2 и 4 исключили возможность использования этого протокола. Представьте 10 бортов должны циклично каждые три секунды передавать данные в которых кодировано 6 параметров. При этом они не должны мешать друг другу.

Изучено было много материала. Камнем преткновения стал протокол передачи данных. Конечно, я много смотрел в сторону AX25. Но как по нормальному совместить передачу данных через радиоэфир на малой мощности, от постоянно движущегося объекта, на условно большое расстояние, да так чтоб гарантированно, при этом объектов сразу 10. И чтоб данные отображались действительные - актуальность 4 секунды?

Пришло понимание того что передаваемый пакет должен быть коротким. Исходя из п. 10 и 5 получается:

4сек / 10бортов = 400мсек 1борт

Пункт 12 так же определил выбор используемых частот - 433 или 144 МГц.

Передать необходимо:

1. МЕСТО с определённой точностью - это данные GPS широта и долгота. Полный передаваемый параметр системой GPS — это два 9-тизначных числа с плавающей запятой!!!
2. ВЫСОТА с определённой точностью. Полный параметр, передаваемый системой GPS 6 знаков
3. Скорость. Закладываю 3 знака
4. Курс. Так же 3 знака
5. Скороподъёмность. 3 знака
6. Номер борта. 2 знака
7. Незабываем что необходимо хоть какое-то избыточное кодирование, плюс преамбулы начала передачи пакета, контрольные суммы в конце передачи. Назовём это «Транспорт» + 50% ресурса – это примерно 15 знаков сверху.

В итоге что имеем: Необходимо кодировать и передавать в эфир 50 байт, на частоте (433) 144МГц малой мощностью, со скоростью 400 мсек (примерная скорость 120 байт в секунду).

На помощь пришёл его величество случай. Как-то в один из зимних вечеров, пришёл ко мне товарищ и принёс мне простенькую самодельную метеостанцию, сделанную на ESP1866. Платка ESP и датчик BME280 всё без корпуса. Просто незатейливо, но всё работало. Идея со сбором метеоданных с того момента начала развиваться. Хотелось сделать больше, собирать больше различных данных. Явно не хватало анемометра и уровня осадков. Стал копать в этом направлении что есть. Перебрав опять же кучу материала, набрёл на статью об использовании таких датчиков - OREGON SCIENTIFIC WRT810 и PCR800. И надо же, эти датчики есть в достаточном количестве в продаже, и даже не дорого! Заказал сразу два комплекта себе и товарищу :). Но вот дальше интересно. Проблема в том, что к ним нет базы. Они беспроводные, работают по радиоканалу 433 MHz. Нужно было строить. В итоге был построен радио-модуль для приёма данных с этих датчиков, но самое главное не это, а то что для этих датчиков не было поддержки в коде. На просторах интернет наткнулся на пост автора Invandy, в котором он описывает работу различных коммерческих датчиков, а также какие-то свои собственные разработки датчиков. На мой взгляд, это лучшая статья, и лучшая любительская работа. Ему спасибо! Начал изучать, и как водится допиливать этот проект под свои задачи. Всё получилось. Датчики были встроены успешно в проект и по сей день работают. Увидеть результат можно на narodmon.ru ссылка на проект git-lira.net. Результат впечатлил. Меня тут осенило. Стало понятно то, что идея применима к тому что я искал столько времени. Просто, дёшево и сердито. Главное повторяемо!

Итак, за основу был взят код из проекта метеостанции INVANDI. Существенно переработанный. Но за идею ему ещё раз спасибо! Общая концепция проекта приобрела вот такой вид:

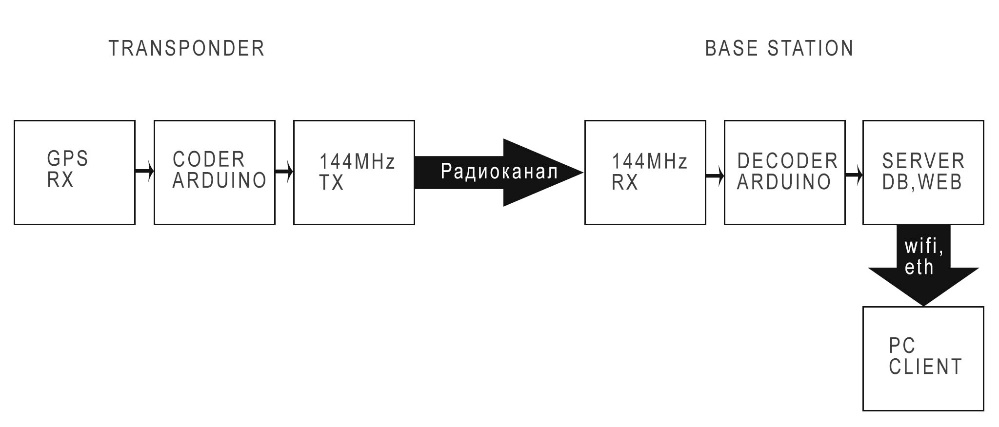


Рисунок Блок-схема проекта.

Как вы понимаете, исходя из схемы, транспондер только передаёт, а БС только слушает. Между собой они не договариваются. Это минус, но зато время передачи пакета существенно сокращается, да и потом, становится ниже вероятность того, что кто-то кого-то перестал слышать и в итоге пакет либо не дошёл, либо был отвергнут из-за квитанции… Остаётся только обрабатывать возможные ошибки на стороне БС и проверять контрольные суммы приходящих пакетов. На стороне БС ресурс производительности условно можно и не жалеть.

## Формирование пакета.

Ну что же всё есть. Вооружившись начальными данными, и небольшим опытом обрисуем схему формирования пакетов.

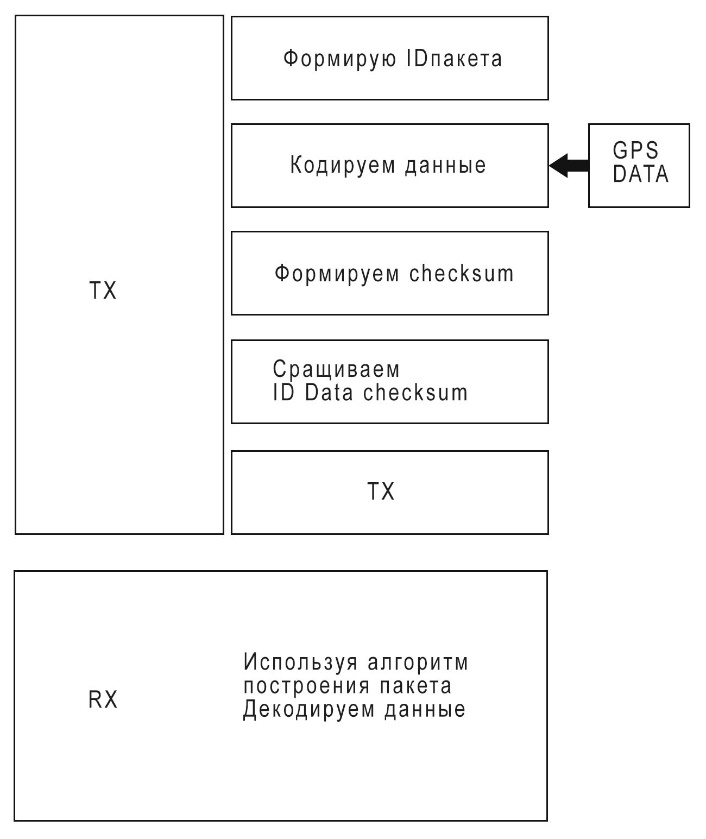


Рисунок Формирование пакета.

Следующее что можно было оптимизировать это код самого транспондера. Слишком велико значение lan и lon получаемые приёмником GPS. Для нашей специфики полётов такая высокая точность просто избыточна. Поэтому принято решения сократить кодируемые координаты до 3х знаков после запятой, да и саму запятую необходимо бы убрать избавившись тем самым от неудобного числа с плавающей запятой. Сократили lat lon до 10 символов, помимо символов «+» и « – » их кодировать достаточно одним битом 1/0 .

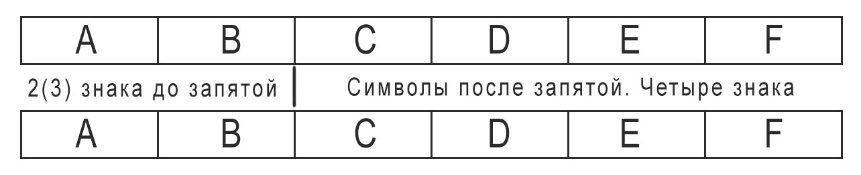


Рисунок Формирование данных lat, lon

Подобные оптимизации касаются и других переменных значений. Подумав еще немного пришёл к выводу, что курс можно не передавать, а рассчитывать его математически на стороне БС, то же самое и сделать с курсовой скоростью, ибо скорость в потоке всё равно взять в Бланике просто нечем. Итак, получилась вот такая расчётная схема кодирования разрядов:

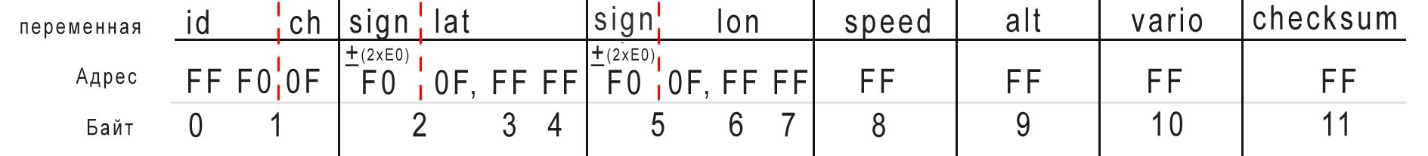


Рисунок Схема пакета V2.

Можно сэкономить байт если использовать пустую часть нибла в переменной lat. В этом начальном нибле можно хранить флаги значений (+,-) для параметров lat, lon, vario. Использовать можно только последних три MSB.



Рисунок Использование свободного бита.

Основываясь на параметрах полёта планёра Бланик L13 так же ограничил остальные параметры транспондера:

|  |  |  |
| --- | --- | --- |
| Параметр | Предел измерений | Адресация/размер |
| Lat | 179,999 | 0F FFFF |
| Lon | 89,999 | 0F FFFF |
| Speed | 255 km/h | FF |
| Alt | 2550 m  (255x10 - множитель 10) | FF |
| Vario | 15 m | 0F |

Если кодировать знаки + - в конец переменных lat, lon – а именно: использовать по одному биту значения 0, 1 как флаги знаков + -. Пространство alt должно быть FF, так как не хватает места для указания параметра alt. Можно взять ещё один свободный бит из переменной lon.

Тесты и отладки показали то, что можно экономить байтовое пространство, и задействовать все свободные биты которые не использовались бы при заполнении. В таблице параметры lat, lon представлены адресным пространством начиная с 0F Т.к. при максимальном числе, например, lat = 179 я могу использовать всего три разряда этого нибла. Нулевой бит этого нибла(2x80) остаётся незадействованным. По этому я могу использовать его как флаг: 1 это «-» , 0 останется «+». Напомню, что в системе GPS, в зависимости от полушария, значения могут быть и минусовыми.

Так же поступил и с ещё одним параметром var (vario) указание скороподъемности так же требует разных знаков. В данном случае параметр lon ещё более короткий чем lat. И в старшем байте остаются целых два свободных бита (4x08, 4x04), чем я и воспользовался. В итоге родился пакет версии 6.

На заметку: Для передачи координат lat/lon, необходимо было избавиться от запятой, так как тип float занимает большое пространство, что увеличивает размер пакета, тем самым снижая его надёжность, а также снижается общая производительность системы. Поэтому я кодировал значение целого числа отдельно от дроби(мантиссы).



Рисунок Пакет v6.

Ужимал пространство пакета, как только мог. Как можно увидеть из этой схемы, параметры curs и vario так же разделили между собой бит. Параметры alt кодируется значением от 0 до 255. На стороне БС этот параметр умножается на 10. Точность измерения в этом случае +/- 10 метров. Не очень то, но пришлось сэкономить. Параметр vario так же ограничен значением +/- 15 метров секунду. Для планеристов такие показатели во время полёта бывают крайне редко, как правило критические. Например, попал в грозовое облако, а вариометр с пределом всего 10 метров в секунду. Напомню, что полёты в облаках запрещены :).

### Идентификаторы пакета

ID – идентификатор пакета. Содержит версию протокола. Так же ID содержит номер канала(timeslot) адрес 1x0F.

Небольшое пояснение по принципу формирования каналов транспондеров. Использован timeslot – то есть каждому транспондеру задаётся интервал и своё время передачи, в общем цикле равном 4 секундам. Например, первый транспондер начинает передачу на нулевой секунде, продолжительность 0,4 сек. Второй на 0,5-ой секунде, продолжительность 0,4 сек. Третий на 1-ой секунде и т.д…



Рисунок ID пакета v6. Определяется 4-мя ниблами.

Можно начать идентификацию с адресного пространства AAA + номер канала(ch). Тогда получаем:

AAA1 – протокол X AIR1 (канал1)

AAA2 – протокол X AIR2 (канал2)

AAA3 – протокол X AIR3 (канал3)

AAA4 – протокол X AIR4 (канал4)

И т.д..

## Стендовые испытания

Считаю нужным сказать пару слов об стендовых испытаниях, так как ошибки допущенные на этапе разработки, больше помогу понять то, как устроен пакет. В стендовых испытаниях уже учувствовал декодер. Для этих целей поначалу я использовал БС от проекта метеостанции, доработал протоколы, допилил что нужно и, пакеты различных версий я уже мог отлаживать. И так дело дошло до очередных стендовых испытаний. Выявлены следующие проблемы:

Неверно был расшифрован пакет

* Неверно была определена длинна пакета
* Неверно была определена CRC checksum

В итоге оказалось, что длинна пакета составляла 17 ниблов, а это 8 байт и 1 полубайт. Перепутал количество ниблов с длинной – нумерация начинается с нуля. То есть первый нибл под номером НОЛЬ. Кроме этого сократилось пространство для полезной информации из-за неправильно подсчитанной мной длинны checksum. Длинна checksum составляла 5 ниблов, то есть 2,5 байта. Это особенность работы протокола OREGON V3. И Так как я в своей изначальной работе с метеодатчиками имел дело с допиливанием хорошо изученного протокола для PCR810, то результатом такой ошибки стало то, что в пакет не вошло 4 параметра:

* Speed FF
* ALT FF
* VARIO F0
* CURS 0F

Всего три байта.

Частично решено

1. Сократил CRC до одного байта, занявшего пространство FF тем самым высвободил один байт.
2. Увеличил длину пакета на 1 нибл (полбайта). И сдвинул CRC вправо (окончание пакета).

Высвободившееся пространство использовал для параметра alt – 0XFF. Ещё осталось полбайта vario. (При необходимости можно будет сдвинуть CRC ещё на полбайта вправо для параметра curs).

## Пакет версии 7.

Результатом отладки стал пакет версии 7. Всего 19 ниблов. Если считать от 0-го, то последним будет 18-ый нибл.

Вид пакета v7. Пример: AAA10D5C909206170F5

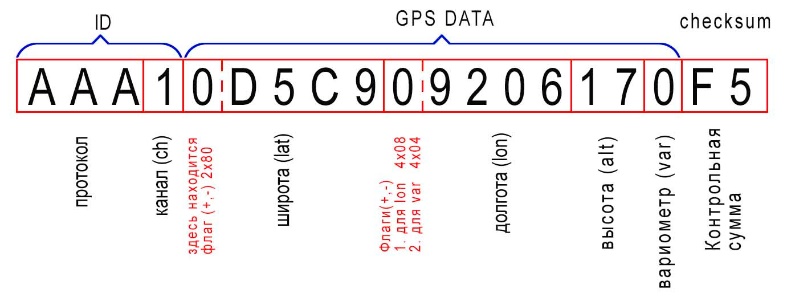


Рисунок Пакет v7

## Полевые испытания

Наконец наступил долгожданный лётный сезон. И можно приступать к полевым испытаниям. Сразу оговорюсь, что зимой-весной, било подготовлено и отлажено всё необходимое железо и ПО, но об этом потом…



Рисунок СКП

Развёрнута базовая станция Старенький Macbook в качестве сервера. Приёмник - их было два, тестировались по очереди: Radoshack и YAESU FT-11r В качестве преобразователя уровней двухканальный ОУ от оптического привода. Декодер – их тоже было два: ESP8266 для беспроводного соединения с сервером, и Arduino UNO с ETHERNET шильдом для проводного соединения с сервером. Вариант на ESP8266 мне понравился больше, так как от него не было цифрового шума, и как ни странно обработка пакетов на ESP8266 была быстрее! Для беспроводного соединения рабочих станций с сервером, использовался WIFI роутер TP-LINK. Ну и в качестве рабочей станции – ноутбук, или смартфон, да что под рукой было…:) Да и самое главное – антенна. Внешняя двух диапазонная антенна OMNI 144/433 MHz, с круговой диаграммой, и с коэффициентом усиления, по паспорту, 3,5/4 db.



Рисунок Транспондер на борту Бланик L13

Транспондер был установлен на один из планёров. Кодер на Arduino Nano. В качестве трансмиттера YAESU FT-11r. Антенна была штатная резинка. На фотографии справа можно увидеть батарею, FT-11r, и беленькая коробочка. Вот оно в таком виде отлетало. Обзору и пилотированию не мешало, приём был хорошим :).

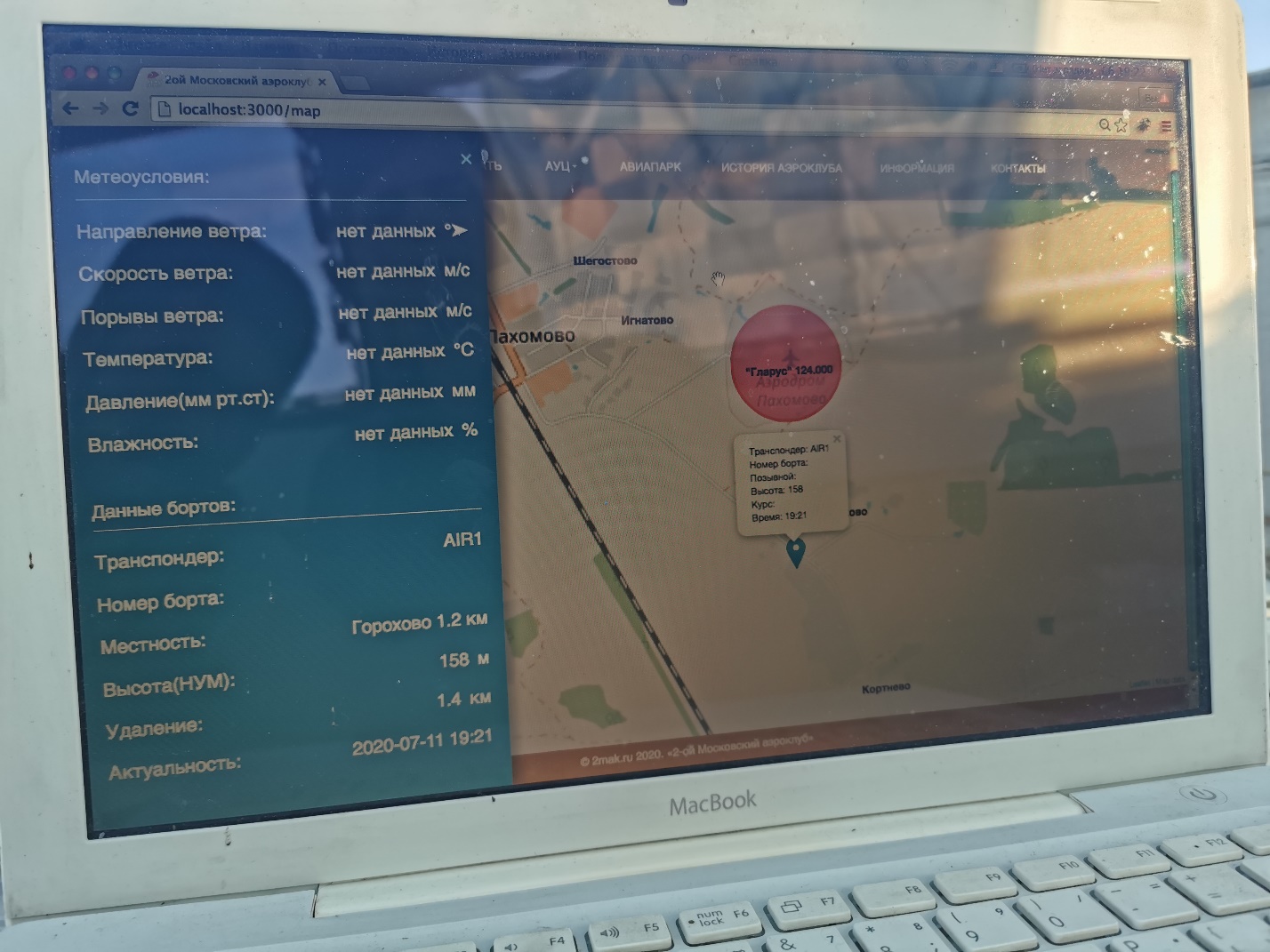
В результате полевых испытаний была обнаружена ошибка передачи параметра alt. При достижении высоты 255 метров, счётчик (указатель высоты), обнулялся. После чего отсчёт вновь продолжался с нуля. И так каждый раз до 255 и вновь сброс счётчика…

Ошибка была на стороне приёмника (в библиотеке get\_altitude) были установлены неверные типы данных – byte. Для корректного приёма этого было достаточно, но для дальнейшей передачи параметров больше 255, нужно было увеличить полученное число на один разряд путём перемножения полученного значения на 10. В результате разрядность числа увеличивалась, превышая размеры указанного типа данных – byte (всего 8 бит), не вмещающиеся в 8 бит старшие разряды отсекались. Так счётчик обнулялся через каждые 255 метров.

Решение было таким: Изменён тип данных всей библиотеки get\_altitude до 16 бит – тип данных word.

## Итоги полевых испытаний

Итоги первых полевых испытаний порадовали. При излучаемой мощности 0.1 Ватт (не опечатка, да всего 100 милливатт) уверенны приём был в радиусе 8 км от точки. Жаль в те дни не было парящих маршрутов. Ходили вокруг да около старта. От экрана я не отрывался. Следил за всеми перемещениями метки.



В интерфейсе, если обратили внимание, я сделал так же метеосводку. Эти данные берутся с тех самых датчиков OREGON и BME280 которые устанавливались так же на аэродроме. На этой фотографии данные отсутствуют, но все датчики я так же оттестировал в полевых испытаниях, тут проблем не было.