Assignment 4: Automatic Dependent Surveillance-Broadcast (ADS-B)

Overview

Last week we listened to airband signals, and pilots talking to air traffic control using AM modulation. This week we'll look at another way of tracking planes using digital packets. This is a nice example of on-off keying (OOK) that you talked about during the last class, and shows how digital data can be transmitted on an analog RF waveforms. Lots of devices talk to each other with this sort of approach.

Introduction

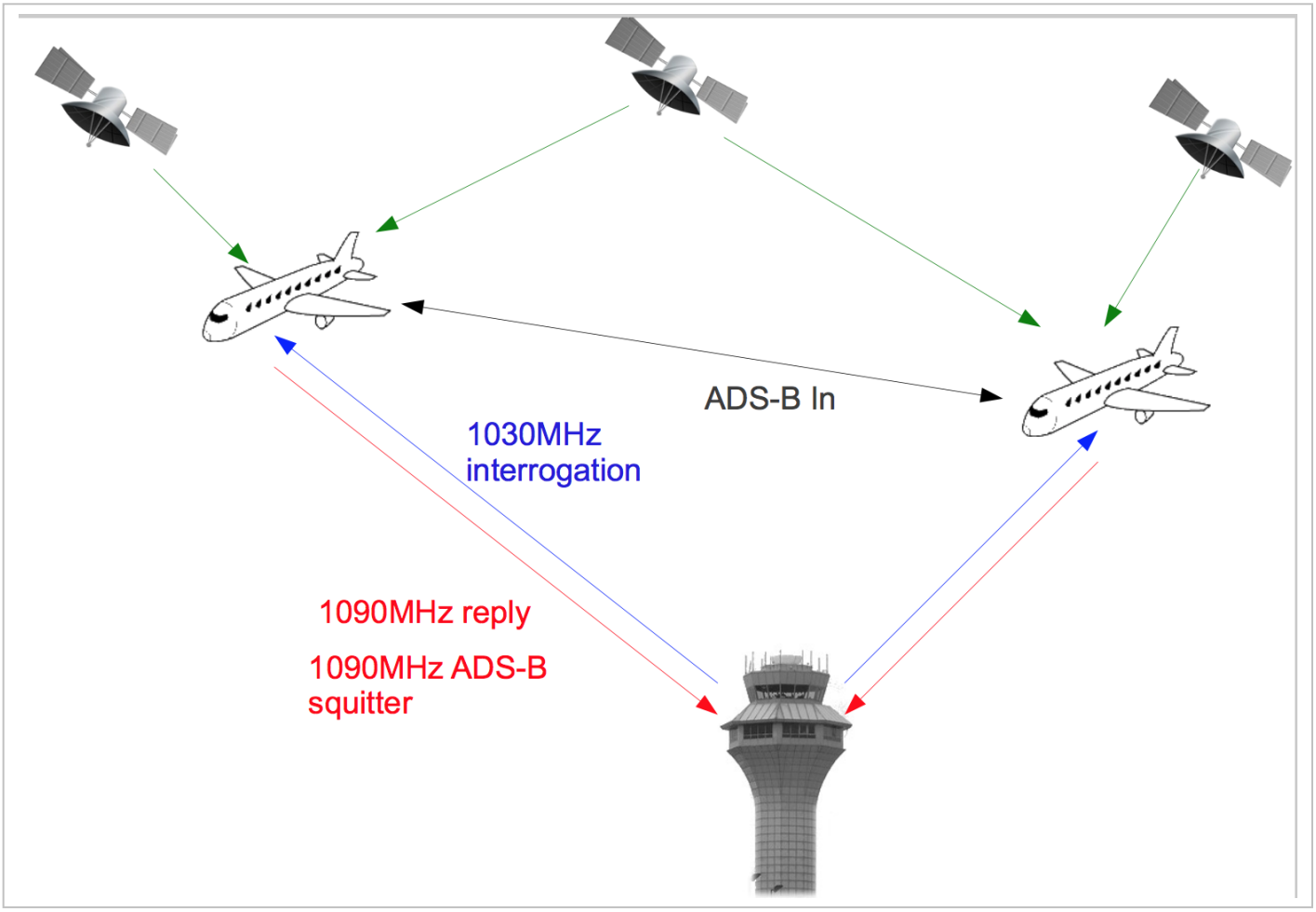
The key element in making air travel safe is keeping track of where all of the planes are and where they are going. In the past planes were tracked with radar. This does a great job of detecting planes, and measuring their range and velocity towards the radar. It does less well for localizing the plane in angle space, due to the beam width, and has no idea which plane it is interrogating. It is also very expensive to keep all of this hardware up and running constantly, for decades.

This system of radars is being replaced with a new system called Automatic Dependent Surveillance-Broadcast, or ADS-B. This is based on transmitters that are currently carried by all planes in the US. Commercial aircraft use these packets to periodically broadcast their tail number, flight, altitude, direction, and speed. Light planes can transmit shorter packets that simply identify them. By 2020 all planes in US airspace were to have had ADSB transmitters to broadcast their full information and status. This didn't quite work out as you will find out, but is getting closer. Other countries have sightly later schedules.

The basic idea of ADSB is that planes have a very good idea of where they are from GPS, and who they are, and where they are going. All the air traffic control system needs to do is ask, and the plane will be able provide all this and more.

The goal is to be able to shut down the radar tracking system at some point in the near future and rely completely on ADS-B.

The general layout of the system looks like this:

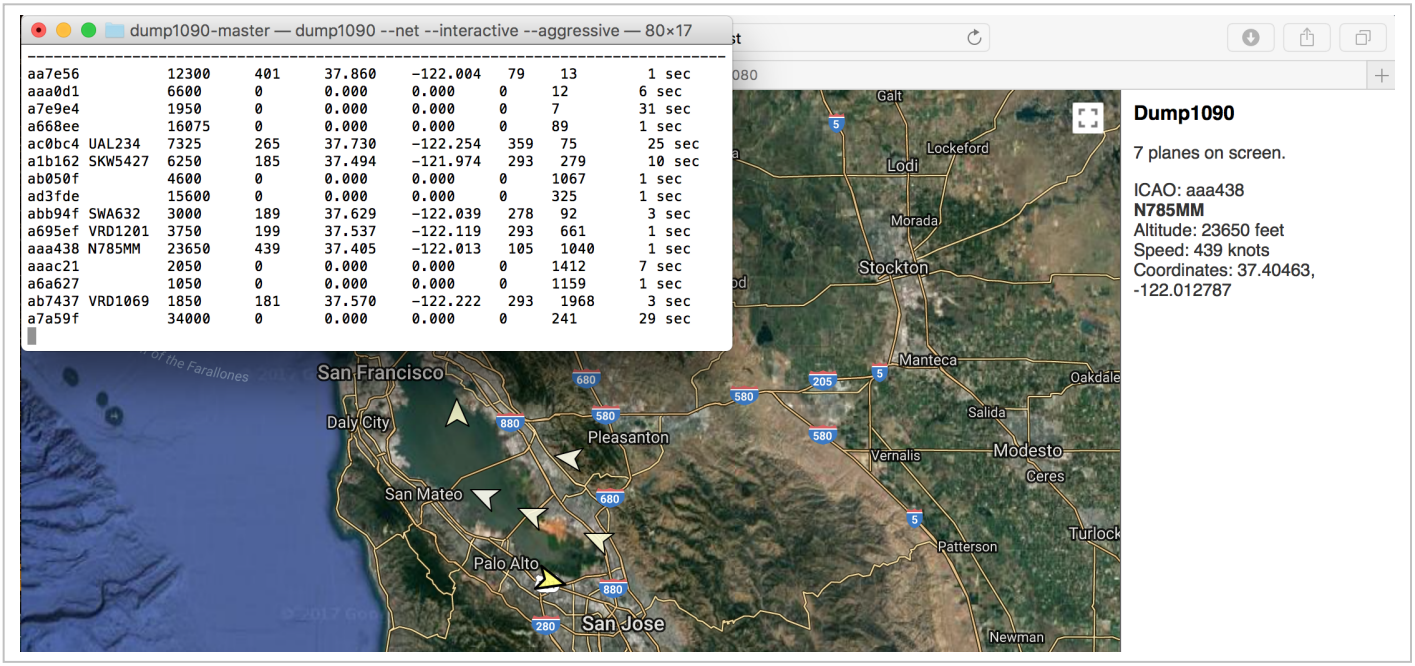


The planes listen to the GPS system to figure out where they are, and then transmit that information to other planes and ground control using short digital packets.

When the system was designed, these radios were expensive (~$10,000). It was expected that only people who really needed to know about aircraft locations would spend the money to buy the radios. Who else would bother? In fact, it turned out the your rtl-sdr and a raspberry pi (or your laptop) can do this for $50. Lots of people set up ADSB receivers, and aircraft position data became an easily available public commodity. This was not what the designers expected.

Since ADSB is not encrypted. The location and identity of every plane is public information. This is significant. For example, numerous law enforcement agencies use surveillance planes or helicopters, and these all have tail numbers, and transmit on ADSB. All of these activities are now publicly available. Google FBI and ADSB, or CIA and ADSB for example. You will find lots of interesting things.

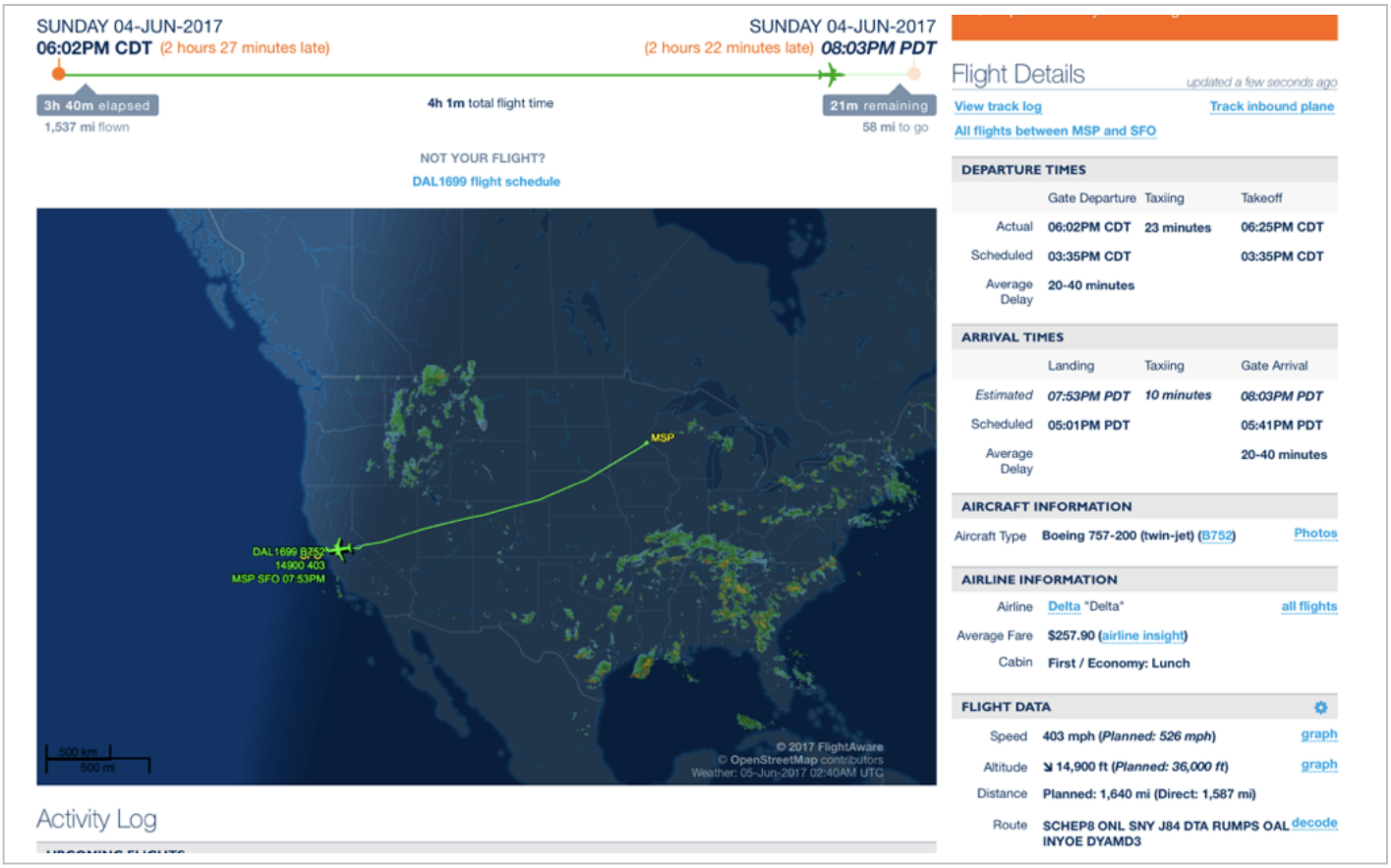
The packets that are sent are called “squitters”, a term which originated in the military with Identification Friend or Foe (IFF) systems. Programs like dump1090 on MacOS and linux (and many more on Windows) use your rtl-sdr to acquire and track the ADSB packets. An example is shown below,



This is in the morning about 9:30. You can clearly see how the planes are lining up for SFO. If you click on a plane, it gives you its flight information.

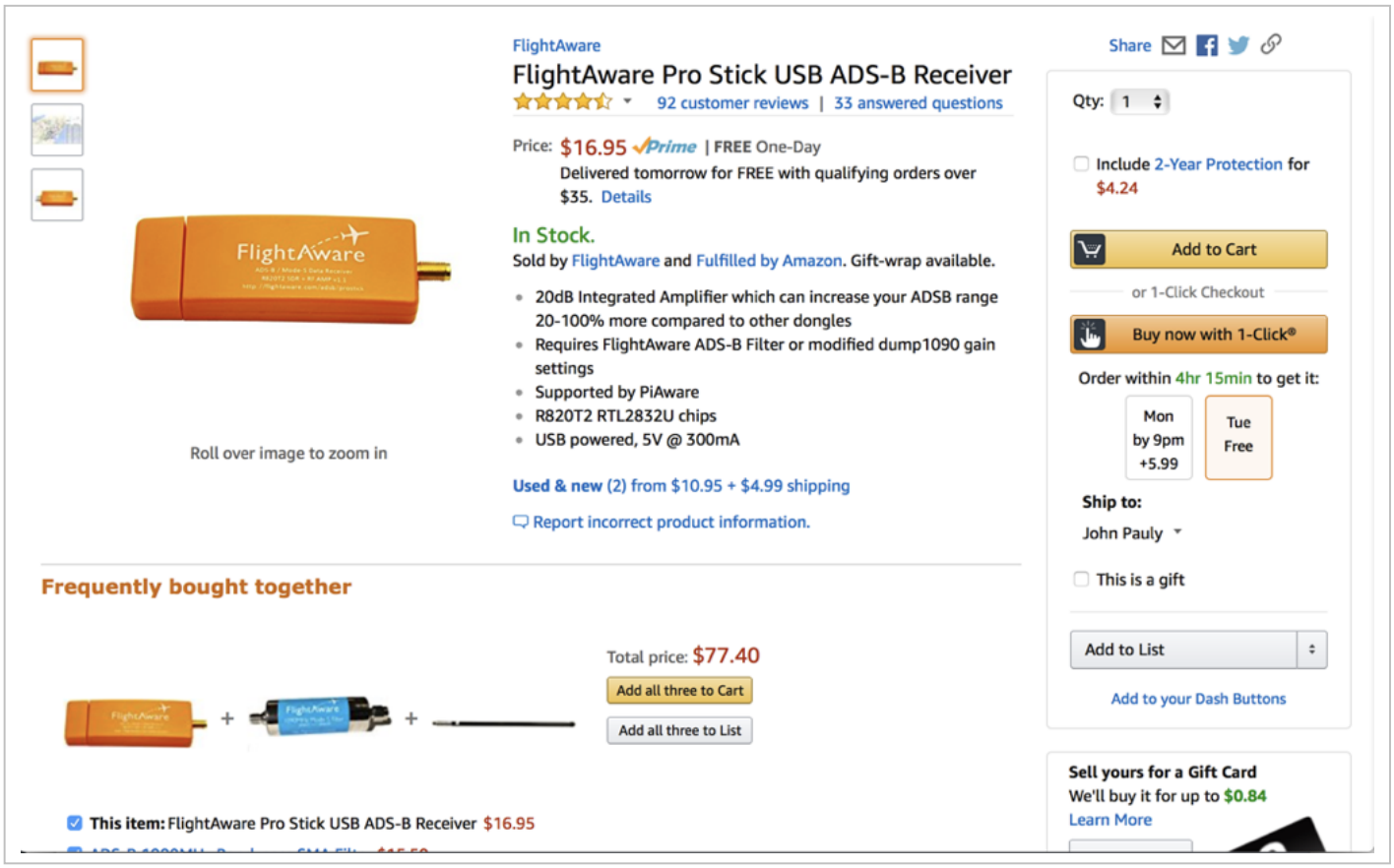
Since ADSB is public, many web sites aggregate ADSB data from a wide range of sources to provide real time information about where flights are, and the flight paths they have taken. Two commercial sites are Flightaware.com and Flightradar24.com. Then there is a public domain data aggregator called adsbexchange.com. These are very useful.

An example from Flightaware for a flight several summers ago is



My wife was coming in from Minneapolis, so I could follow every twist and turn of the flight.

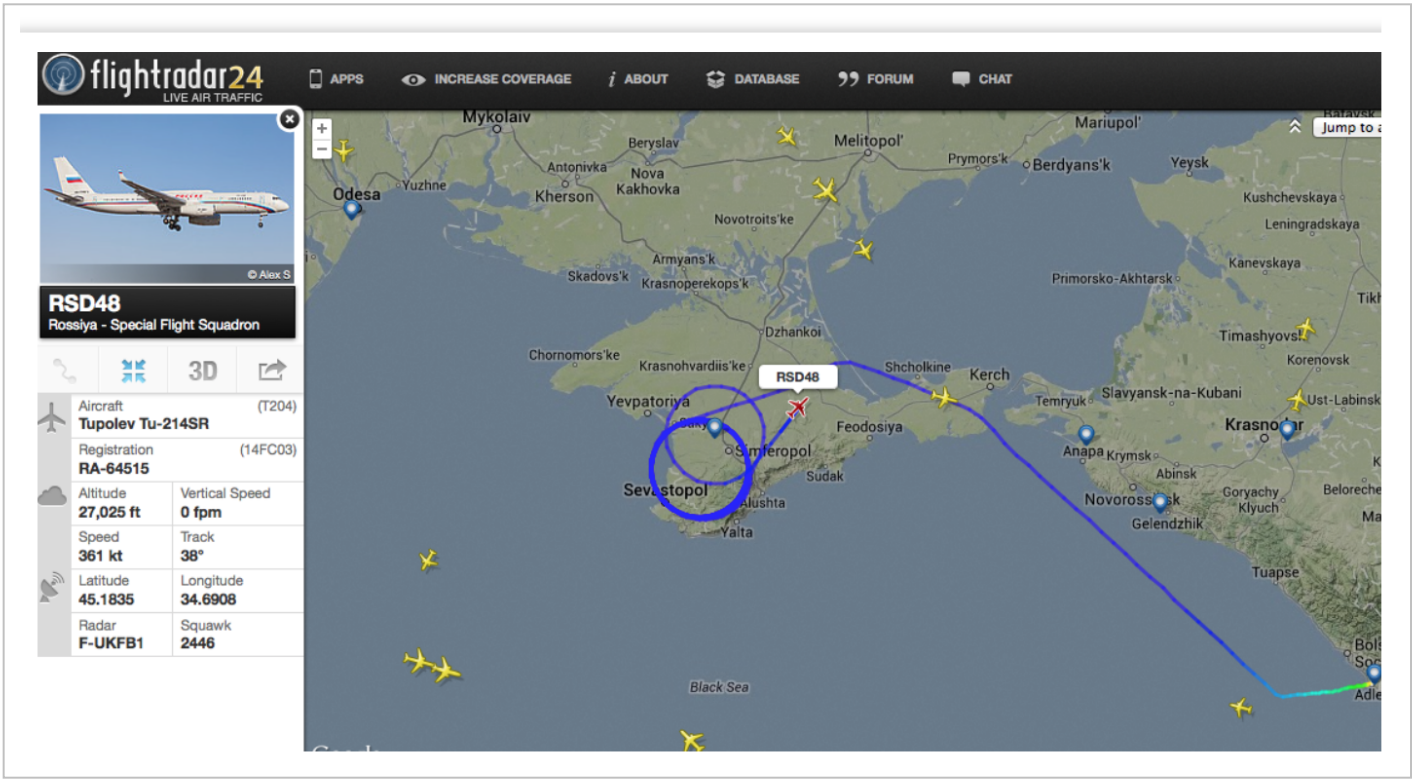
These sites actively recruit people with rtl-sdr's to participate in their network, and even sell branded rtl-sdr's explicitly for this purpose:



The hardware is free if you are in an area that needs coverage (sorry, the Bay Area is already pretty well saturated!). The software runs on a Raspberry Pi. Lots of people set up nodes, and forward information to these services.

You can also follow other people and their flights. Vladamir Putin's aircraft transmits on ADSB, and you can track his trajectory. One plane he often uses has a tail number RA-96012. If you type this in to flightradar24.com, it will show you where he has flown in the last week. Pay for a subscription, and you can follow his flight for the last couple of years. It is pretty interesting which airspace he flies through, and which airspace he avoids!

Another interesting example is Putin's doomsday aircraft. This is a telecommunications aircraft that circles above his location wherever he is, to make sure that if buttons need pressing, they will work. There are a couple of these, but one example is here

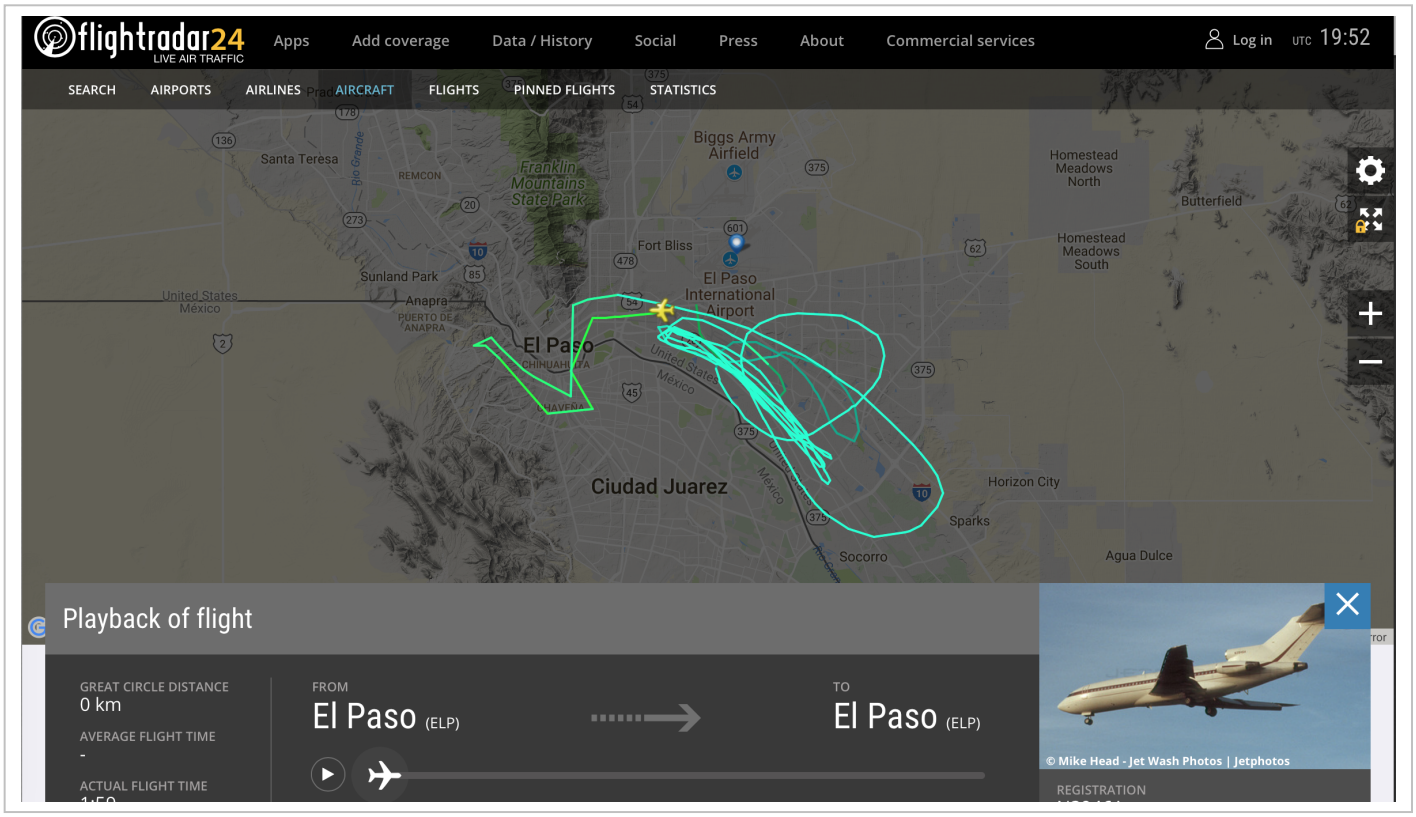


For the U.S., Air Force 1 generally turns off its ADSB transponder, but occasionally does turn it on. Closer to home, there are lots of interesting aircraft flying around the U.S. The web site

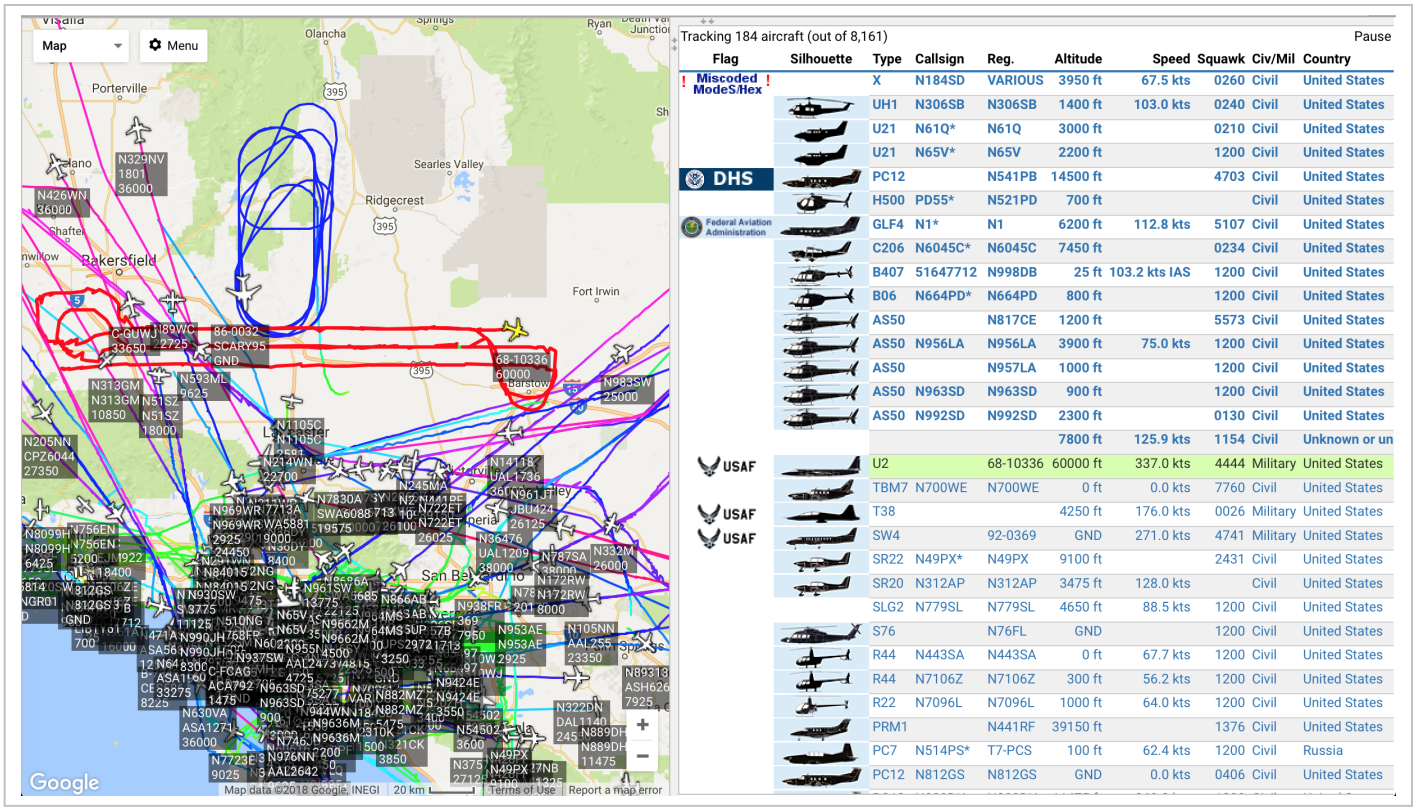
Interesting Aircraft

is a forum where people point out some recent candidates. The commercial sites filter out some "sensitive" aircraft. Adsbexchange.com does not.

Some of the interesting aircraft also show up on the commercial sites. Here is a Department of Homeland Security plane flying out of El Paso:



If you go directly to the adsbexchange.com site, and click the “global radar view” tab, you will be able to see everything that is in the air now. One recent afternoon in Southern California looked like this



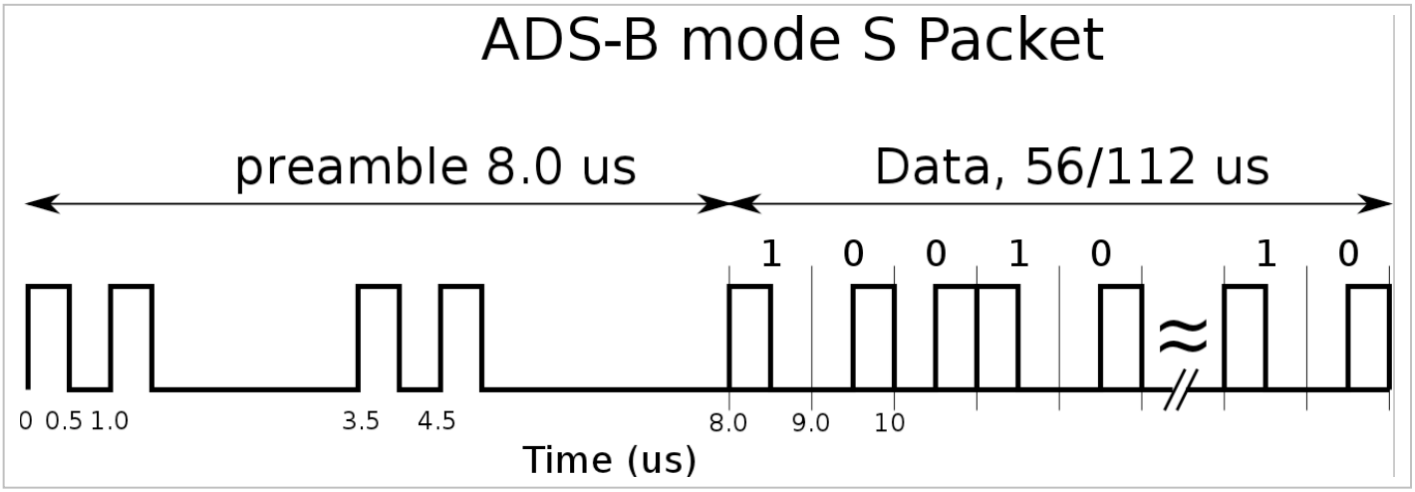
We see a U2 rastering back and forth across Edwards Air Force Base, with a tanker holding in a loop to the north. If we shift to the west, an E2C Hawkeye is operating just off the coast of Point Magu Naval Air Station. A DHS plane is circling over Pomona to to south.

This is a degree of transparency that was not what the designers intended. The question is what to do next!

Before we get to that, lets first look at the technology behind all of these issues.

ADSB Packets

The ADSB packets are short sequences of pulses, that look like this:



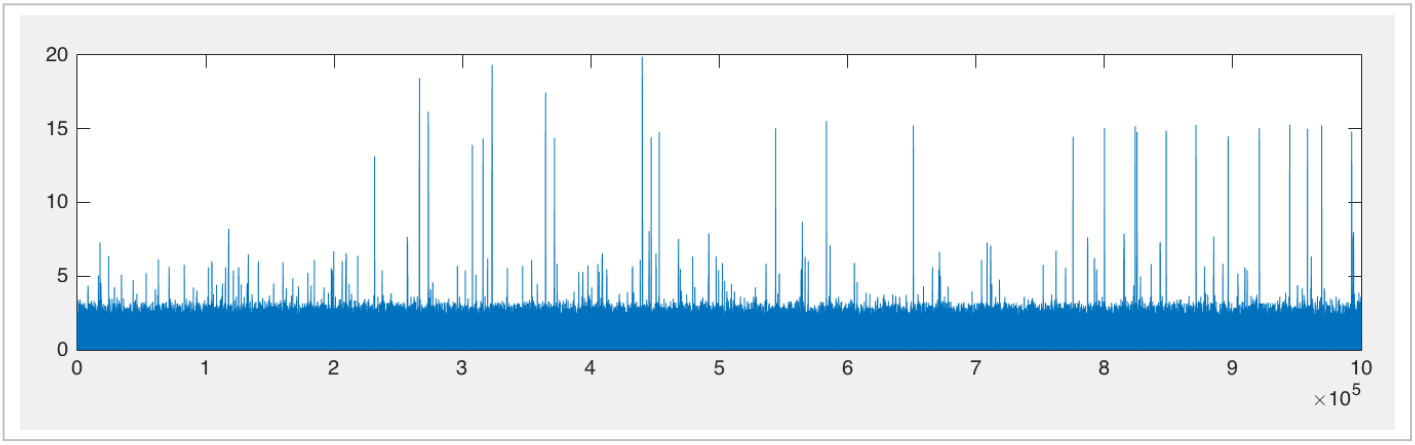
Each pulse is 1 us long, and consists of two 0.5 us subpulses. This is an example of on-off keying (OOK) just like you described on Tuesday.

There is an 8 us preamble that allows you to find the start of the packet. It's spacing is unique, and can't turn up in a packet. When you see that pattern, you know you have a packet, and exactly where it starts. This is important to make sure it decodes properly.

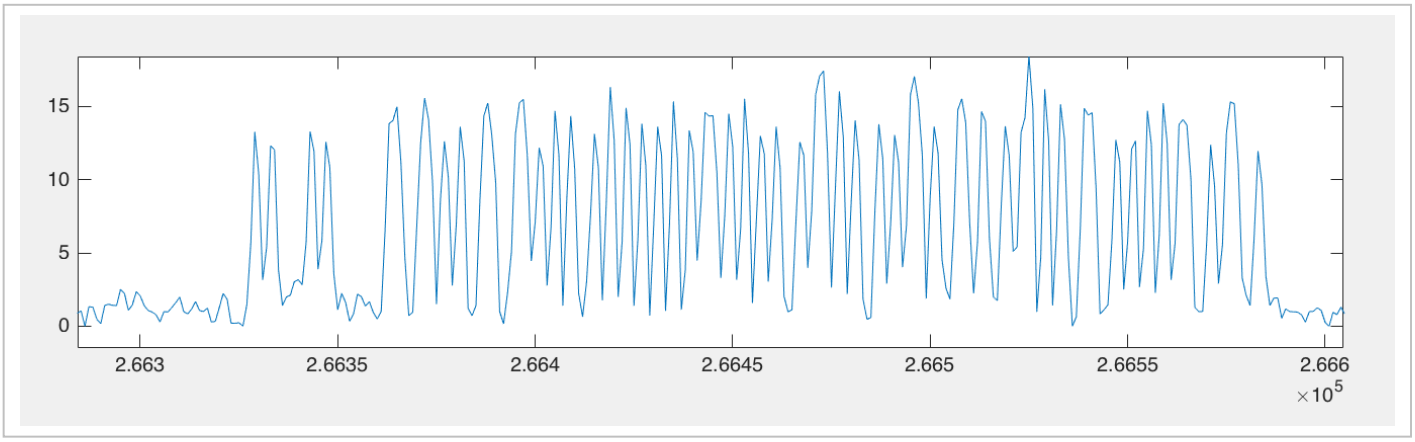
The packet itself follows, and is either 56 or 112 us long. This decodes to 56 or 112 bits. Bit are encoded as split phase pulses, where a falling transition is a 1, and a rising transition is a 0. Every bit has a transition, which makes it easier to keep the timing of the bits right. If we just used ones and zeros, we could lose timing if there was a long sequence of either just ones or just zeros. The split phase pulses fixes this problem.

The ADSB signal is transmitted at a frequency of 1090 MHz. Everyone talks on the same frequency. The packets can interfere. However, since there is a large number of distributed receivers, each signal will be strongest somewhere, and most packets will get received.

If we capture the raw RF of the ADSB signal and plot the first 1e6 samples (half a second), we see something like this

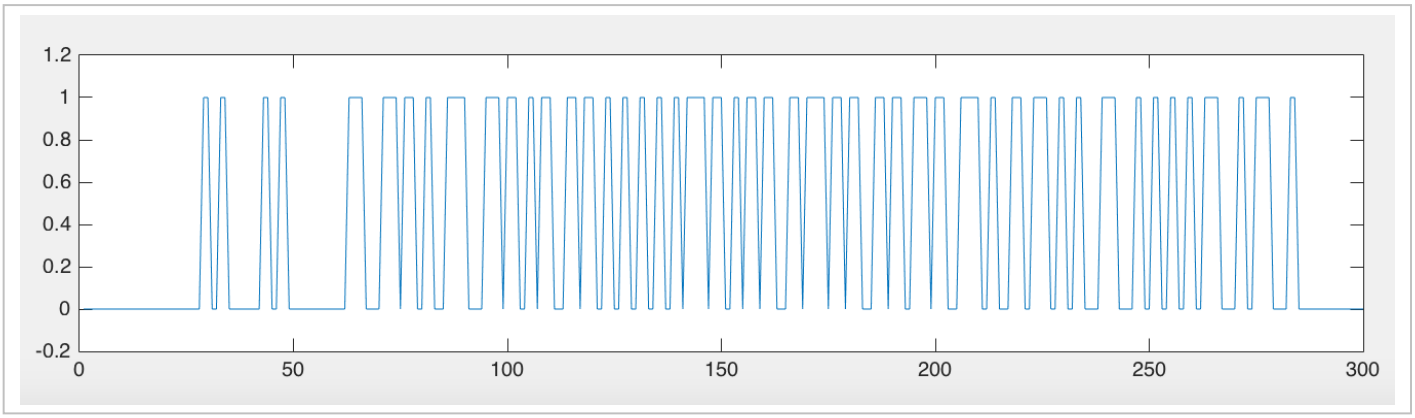


Each of the spikes is a packet. If we zoom in on one packet, we see something like this

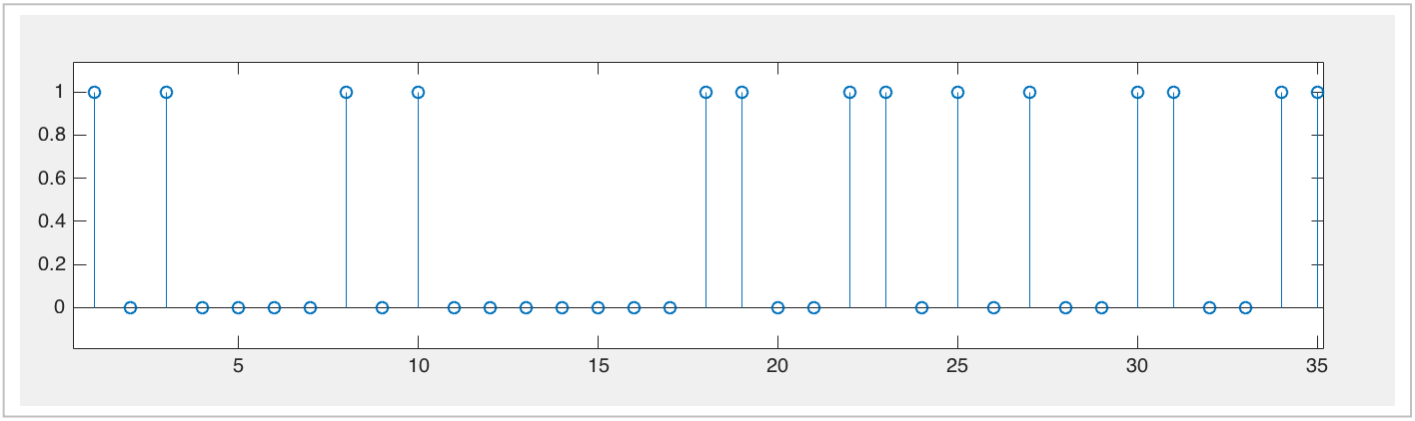


You can clearly see the preamble and the data bits.

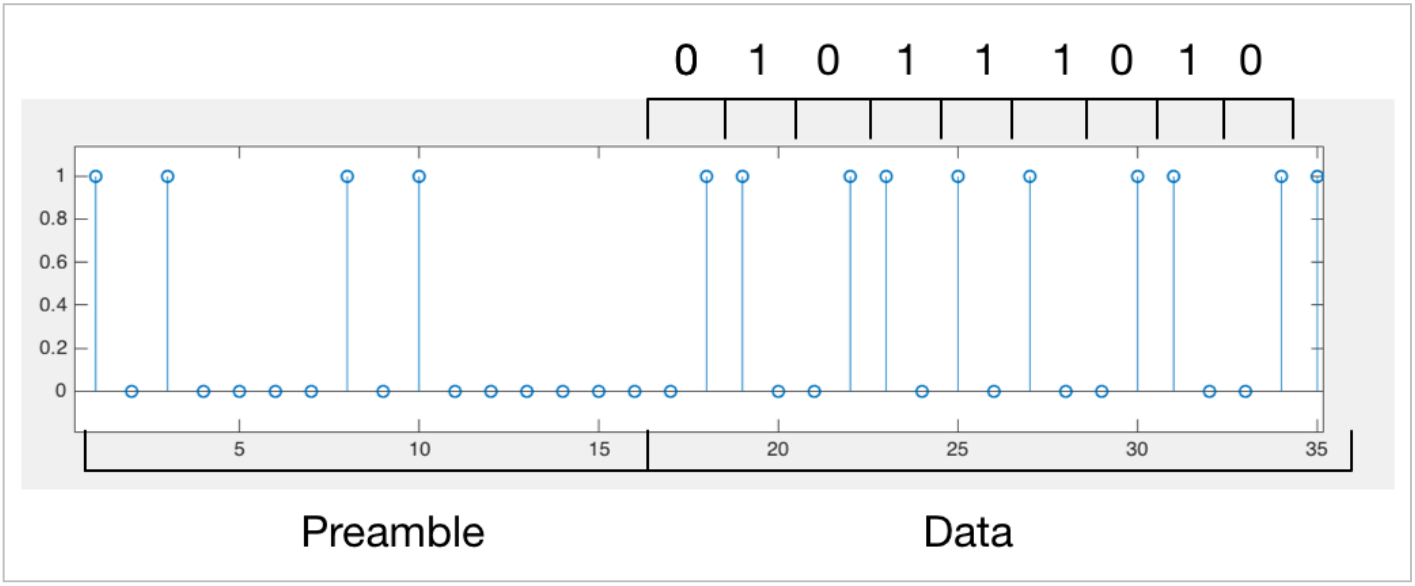
If we threshold the analog waveform we get the digital ADSB waveform



Sampling once per 0.5 us, we get the ADSB bit stream for the packet.



We decode this as



Since the packets are only 56 or 112 bits, a lot of effort has been devoted to compressing as much information into the packets as possible. This is fairly intricate. We'll restrict our attention to packets with the fight numbers of the planes. Each plane sends these out every 2 s.

The bits in the 112 us ADSB packet are allocated as follows,

nBits	Bits	Abbr.	Name
5	1 - 5	DF	Downlink Format (17 or 18)
3	6 - 8	CA	Capability (additional identifier)
24	9- 32	ICAO	ICAO aircraft address
56	33 - 88	DATA	Data
	[33 - 37]	[TC]	Type code
24	89 - 112	PI	Parity/Interrogator ID

The first 5 bits after the preamble identify the type of packet. This will be 17 or 18 for planes communicating to the ground. We are only interested in the DF 17 packets. The next three bits tell something about where the plane is, we'll skip these.

The unique identifier for the plane is the next 24 bits. If you format these bits as a hexidecimal string, and type that into google, you will find out which specific airframe this is, who owns it, and what they do with it.

After that is the data bits. This looks like

```
     DF--- CA-  ICAO--  DATA-----------------  PI----
HEX: 8   D      4840D6  2   0     2CC371C32CE0  576098
BIN: 10001|101  ******  00100|000 ************  ******
DEC: 17   |4            4     0
                        TC    *
```

The data field can be decoded for the flight number like this

```
HEX: 20         2CC371C32CE0
BIN: 00100000 | 001011 001100 001101 110001 110000 110010 110011 100000
DEC:          |   11     12     13     49     48     50     51     32
LTR:          |   K      L      M      1      0      2      3      _
```

One of the packets contains the flight number for the plane. This is sent out every 2 s. This is the identify packet. These have an initial 5 bits (bits 33-37) of [0 0 1 0 0], like the packet above. This is the TC field.

After the initial 8 bits of the data packet, the characters for the flight number are sent as six bit integers, where each integer encodes for a character

```
#ABCDEFGHIJKLMNOPQRSTUVWXYZ#####_###############0123456789######
```

The # entries are not used. This can be summarized as 1-26 => A-Z, 48-57=> 0-9, and 32 => space.

This is a pretty typical of packet data. Things are pretty simple in the analog RF domain that is used to transmit the bits. Once the bits are decoded things get complex. Lots of information is crammed into as little space as possible. Prefix fields completely change how the rest of the packet is decoded. Packets can be hard to decipher if you don't have an explicit specification available.

Assignment

You have several options for your assignment this week. For each topic, generate about 5 slides to describe your thoughts or results. Sign up here

Week 4 Signup

Upload your slides here:

Week 4 Slides

1. Security and Reliability

So we're going to shut down the radars, and rely on planes to tell us where they are. What can possibly go wrong with that? Some things to think about are security, reliablility, and expandability.

One interesting issue is what happens when there are too many planes, and the spectrum gets saturated. This is already an issue at big airports, and could be a much bigger problem with flying cars, and putting ADSB on drones.

Another interesting apsect is that no security is incorporated in the ADSB protocol. The packets are not encrypted, there is no way to tell if they were sent by the plane they claim to be from, and there is no way for the plane to tell if the packet was received. What are some of the issues this raises?

2. The Military

The military is unhappy with exposing all of their aircraft to public tracking. They have been trying since 2010 to find a workable solution, with some progress. Find out where the military is now, and what their plans are.

3. ElonJet

Another recent issue was the ElonJet web site/Twitter feed. As you can now tell, this is really not hard to implement. What was the controversy? In the last two weeks, ADSBExchange was bought by a commercial company, which makes them vulnerable to pressure or outright acquisition. The people who run the data feeds are concerned. Describe what is happening here. The ADSB Reddit thread is a good place to start.

4. Interesting Aircraft

There are lots of interesting aircraft out there. Use the “Interesting Aircraft” link above and track down a couple planes. Who is using them, what are they are doing, and what do their flight tracks look like?

One area of the world to look at is Ukraine. What do you see there for aircraft traffic? If you click the “U” button on the upper right of the map, it will just show you military aircraft. You wont’ see Russian or Ukrainian military planes, but you may see tankers, transports, and surveilience aircraft from NATO, Turkey, and the US.

5. Dictator Alert

There are lots of places in the world where people (particularly governments) are unhappy with ADSB. One interesting web site that has drawn a lot of attention is Dictator Alert, which was set up by some investigative journalists. What is their story? Another interesting event concerned a UN Official visiting Tunisia. See what you can find out about that.

6. Acquiring ADSB Yourself

Use your rtl-sdr to acquire ADSB data directly. Set your antenna up with an appropriate length, and vertical polarization. For Windows users the most popular program seems to be Plane Plotter. This comes as a self extracting .exe file. It has a free 21 day trial period, after which it costs 25 Euros. For the Mac and Linux, the key piece of software is Dump1090. This captures and decodes the data. The actual display is usually handled by other programs (typically a web server and a browser). Some basic installation guidelines for the mac are here. Raspbian even has dump1090 in its package manager. 1090 MHz doesn't go through buildings too well, so put your antenna in a window, or go outside.

How Things are Going

Finally, I'd like to know what you think of the course so far. What do you like, and what should I change. Send me email with your thoughts to pauly@stanford.edu. If you want to be anonymous, have some random friend send the email, I promise not to track you down. Is it too much/not enough work? Is it too technical? Is it interesting or fun? Have you learned how to get yourself into serious trouble with the FCC (you are getting there!)?

Thanks!