

Foreward

My philosophy of learning can be summed up by the following.

When I am told, I forget
When I see, I remember
When I do, I understand
(Confucius)

I first became aware of GPS in late 1992. At that time I was working for Rockwell International and had started working in earnest on the integration of GPS into the B-1B navigation system and the addition of GPS guided weapons in 1993. At about this same time I began to wonder if I could build my own GPS receiver in order to learn in detail their inner workings. At that time I found Dan Doberstein's website at http://www.dkdist.com/articles/Gps_art.html where he showed in detail how to make the front end of a receiver along with a hardwired tracking loop. This was very interesting but I also wanted to decode the navigation message and compute position and velocity solutions.

I eventually found out that GEC Plessy produced a GPS chipset which consisted of a down converter and correlator chips that had 6 channels. Two of these chips could be coupled to produce a 12-channel receiver. In 1995 I laid out a circuit board with the intent of making it functionally equivalent to the GPS Builder that GEC Plessy offered at the time. After many tests by early 1996 the front end appeared to be properly tuned.

Also in 1996 I realized that the self-test wrap-around signal was the best way to test acquisition and tracking algorithms. Along with the hints provided by the GEC Plessy documentation I was able to quickly get one channel to track, record and decode the navigation message, get multiple channels to track and then process the measurements to arrive at a position fix. I guess you could say my time to first fix was about 6 months.

This documentation is meant to provide someone with an interest in the inner workings of a C/A GPS receiver an understanding sufficient to be able to read the source code, to follow what has been done and hopefully to extend it further.

The philosophy of this documentation is to create more opportunity for collaboration by explaining the workings of the software in more detail than I find is practical in the code itself. In each section we start with an explanation of the physical attributes we are dealing with, the mathematical algorithms used, and finally the source code that implements these algorithms. While adding the code produces a large amount of text I believe this is a good way to start but I am certainly open to other suggestions.

When I started the web site in 2001 I was hoping that there was enough interest that this could become a collaborative effort since free time was getting harder to come by as I worked on my Ph.D. at USC. Now I'm happy to say that a number of people have become involved in adding, modifying

the code for different operating systems, adding new functionality, suggesting improvements and especially by asking good questions that require clearly stated answers. I am hopeful that this documentation will encourage even more collaboration

I also believe that the curse of knowledge is that the more one learns the more one realizes how little one knows. While the software appears to work well I find that there are two kinds of software bugs, the kind you know about and the kind you don' t, but there are always bugs.

I will highlight in **red** where I am not happy with what is written or I believe that something is missing. Please feel free to suggest changes and, if anywhere you find the explanation is not clear or you find an error please let me know and your comments and name will be added to the list of contributors. At this point I see my role as an editor and organizer in this Open Source documentation effort.

This is going to truly be a living document. I expect it to be updated often based on comments, question and even some errors.

I would like to thank the following contributors:

Jenna Jingrong
Joel Barnes
Alberto Perez
Doug Baker
Phil Bender
Georg Beyerle
Andrew Greenberg
Elmer Thomas