

UMA DISCUSSÃO SOBRE A UTILIZAÇÃO DO SOFTWARE LIVRE GPL-GPS PARA A PROGRAMAÇÃO DE GPS EM SATÉLITES DE ÓRBITAS BAIXAS

Lorena Gayarre Peña

lorenagayarre@gmail.com

Instituto Nacional de Pesquisas Espaciais- INPE/DMC
P. O. Box 515, 12201-010, São José dos Campos, SP, Brasil

Agustín Martínez Hellín

amartinez@srg.aut.uah.es

Edifício Politécnico Superior, Campus Universitario
Ctra. Madrid Barcelona km 33,600
28871, Alcalá de Henares, España

Marcelo Lopes de Oliveira e Souza

marcelo@dem.inpe.br

Instituto Nacional de Pesquisas Espaciais – INPE/DMC
P. O. Box 515, 12201-010, São José dos Campos, SP, Brasil

Resumo: *Atualmente, o uso de Sistemas Globais de Navegação por Satélite - GNSS tem sido amplamente difundido para várias aplicações, sendo o Sistema de Posicionamento Global - GPS o mais usado. Alguns projetos estão usando GNSS para o controle de atitude dos satélites, seja usando tecnologias por hardware (FPGAs para DSPs) seja por software (microprocessadores). Existe uma larga lista de projetos GPL (General Project License) sobre o sistema GNSS, a grande maioria deles baseados em GPS. Neste trabalho estudamos a possibilidade de programar um sistema GPS para satélites de órbitas baixas (LEO-Low Earth Orbits) utilizando o software livre GPL-GPS de Clifford Kelley. Para tanto: 1-fazemos uma introdução ao sistema GPS (bandas, tempos, frequências de comunicação, etc.); 2- fazemos uma introdução ao projeto GPL-GPS; 3- fazemos uma discussão sobre a utilização do software livre GPL-GPS para a programação de GPS em satélites de órbitas baixas. Esta discussão pretende basear propostas de mudanças para que o software GPL-GPS possa ser utilizado no projeto/designação de sistemas GPS para satélites LEO.*

Palavras-chaves: GNSS, GPS, LEO, software livre (GPL-GPS).

1- Introdução

Neste trabalho estudamos a possibilidade de programar um sistema GPS para satélites de órbitas baixas (*LEO-Low Earth Orbits*) utilizando o Software livre GPL-GPS de Clifford Kelley. O trabalho está organizado assim: na Seção 2 apresentamos o funcionamento do sistema GPS; na Seção 3 fazemos uma introdução ao projeto GPL-GPS e, especificamente, apresentamos o software para GPS de Clifford Kelley; e na Seção 4 discutimos as mudanças que deveriam ser feitas no software de Clifford Kelley para ser utilizado em satélites LEO.

2- O sistema GPS

O GPS (*Global Positioning System*) é um sistema global de navegação por satélites (GNSS), desenvolvido pelo Departamento de Defesa dos Estados Unidos, é capaz de fornecer informação horária e posição de um objeto móvel em qualquer parte do mundo. O sistema GPS está composto por uma constelação de 24 satélites girando ao redor da Terra organizados em 6 órbitas diferentes a uma altitude de 20200 quilômetros como mostra a Figura 1. Desde qualquer ponto da Terra e a qualquer hora é possível ver ao menos 4 destes satélites. Cada um dos satélites tem a bordo um relógio atômico que fornece a hora com grande precisão. Os receptores se sincronizam com os satélites GPS para poder fazer as medidas de distância.

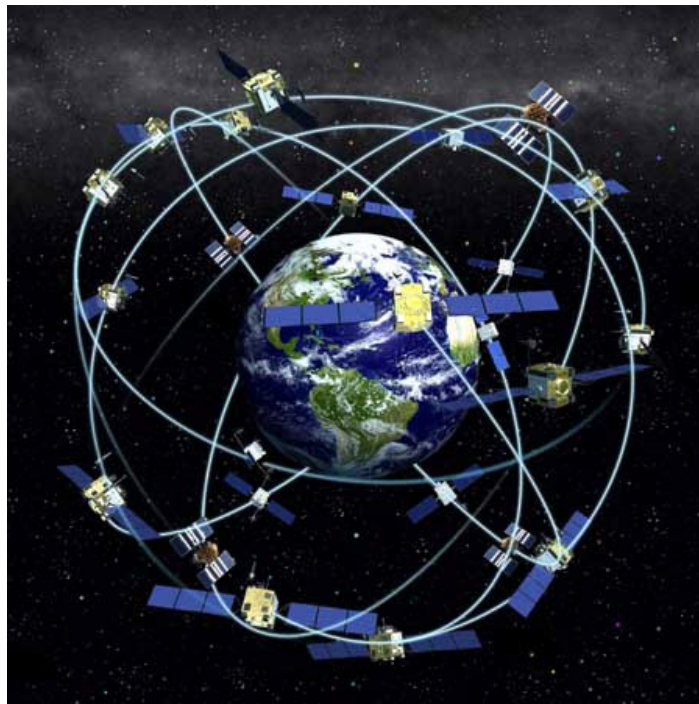


Figura 1: Constelação GPS. Fonte: [Alfaro].

Os satélites GPS transmitem em duas frequências, L1 e L2, à velocidade da luz. Neste artigo vamos estudar só a frequência L1 pois a sinal L2 tem fins militares.

Como todos os satélites GPS utilizam a mesma frequência L1 para transmitir a informação, cada um vai usar um código único (C/A PRN) que permite codificar a informação dele e diferenciá-la dos sinais dos outros satélites. Estes códigos são uma sequência pseudoaleatória de 1023 bits com taxa de bits de 1.023 MHz, e são conhecidos pelos receptores. Sobre este pseudocódigo está modulada a mensagem de navegação (NAV) que contém as informações horária e orbital do satélite. A Figura 2 mostra como é a codificação do sinal GPS.

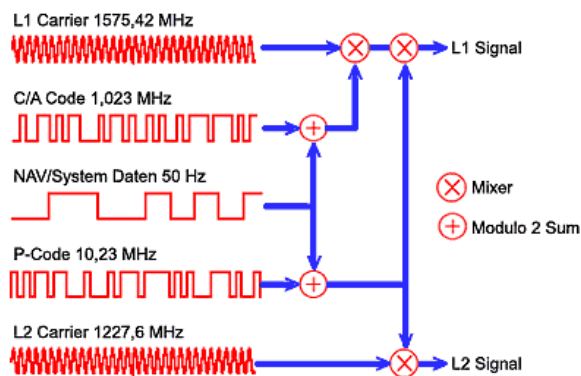


Figura 2: Codificação da mensagem GPS. Fonte: [Alfaro].

Devido ao fato de que os satélites estão se movimentando, o sinal enviado vai sofrer efeito Doppler. Assim, a frequência que chega ao receptor vai estar num intervalo ao redor da frequência L1; este intervalo depende da velocidade relativa entre o satélite e o receptor.

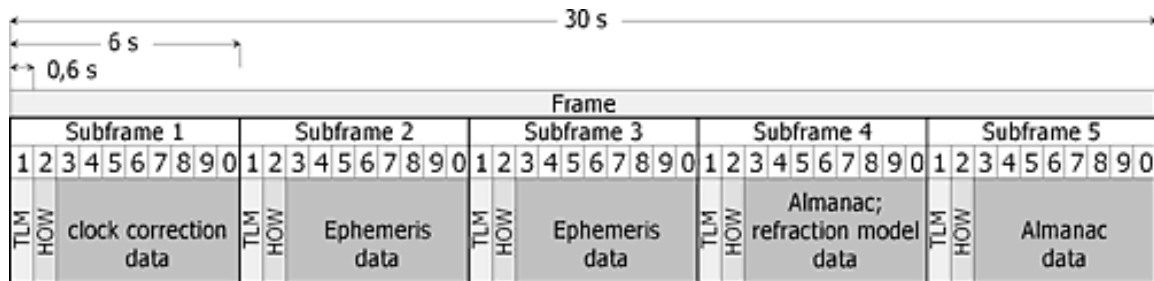


Figura 3: Estrutura do código de navegação dos satélites GPS. Fonte: [Alfaro].

O código de navegação (NAV) está formado por 5 tramas, como se mostra na Figura 3, que contêm as informações de tempo, efemérides (posição do satélite no espaço), e almanaque (estado da constelação GPS) entre outras. O objetivo do nosso receptor é receber o sinal L1 e se sincronizar com ele para ler as informações contidas na mensagem de navegação (NAV).

Quando o sinal chegar ao receptor, este deve decodificá-lo usando o mesmo PRN do satélite que está enviando o sinal na mesma frequência que foi enviado. Cada satélite tem um código PRN distinto e os códigos não são correlatos entre eles para se poder determinar o satélite que foi localizado.

Devido ao efeito Doppler, a frequência recebida no receptor pode não ser a mesma que a frequência real do sinal. É por isso que os receptores precisam de fazer uma varredura em frequências para poder localizar o sinal. Esta varredura depende das velocidades relativas do emissor e do receptor.

O objetivo dos receptores GPS é monitorar ao menos quatro dos sinais destes satélites para calcular a posição mediante triangulação. Existem várias formas de monitorar os sinais GPS. Neste trabalho, o sistema utiliza um banco de doze correlatores. Cada correlator vai integrar o sinal recebido com um código PRN diferente fazendo uma dupla varredura no código PRN e no intervalo de frequências que abarcam o efeito Doppler. A saída do correlator será máxima quando este tenha integrado o sinal recebido com o offset do código PRN e a frequência correta; e zero em qualquer outro caso. O receptor e os satélites estão sincronizados, assim pode-se conhecer o tempo de viagem do sinal fazendo a diferença entre o momento que o sinal saiu do satélite e o momento que ele chegou ao receptor.

Conhecendo o tempo que o sinal demorou em chegar até o receptor e a velocidade de transmissão do sinal (L1 viaja à velocidade da luz) calcula-se a distância (pseudodistância) entre o satélite e o receptor. Com 4 pseudodistâncias pode-se aplicar triangulação para conhecer a posição exata do receptor.

3- O software GPL-GPS de Clifford Kelley

O Software GPL-GPS foi escrito por Clifford Kelley como trabalho de doutorado e os resultados foram apresentados no ION GPS 2002 no artigo “OpenSource GPS: Open Source Software for Learning about GPS” [6]. Ele foi desenhado para a placa Zarlink GP4020 e está programado sobre o sistema operacional de tempo real *eCos*. A estrutura do software está composta por uma máquina de estados que faz o monitoramento do estado dos correlatores e sete tramas (*threads*) que vão decodificar a mensagem de navegação (NAV) calcular as pseudodistâncias e, finalmente, fornecer a posição do receptor. As Figuras 4 e 5 mostram um fluxograma de funcionamento do software e do relógio de controle.

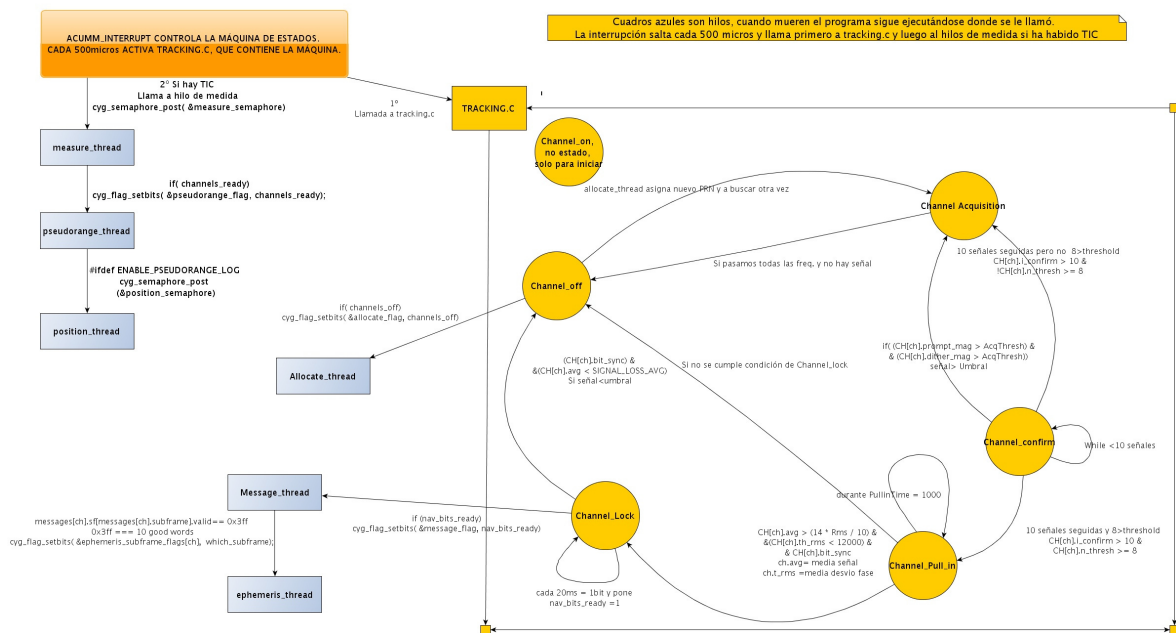


Figura 5. Fluxograma do software GPL-GPS de Clifford Kelley.

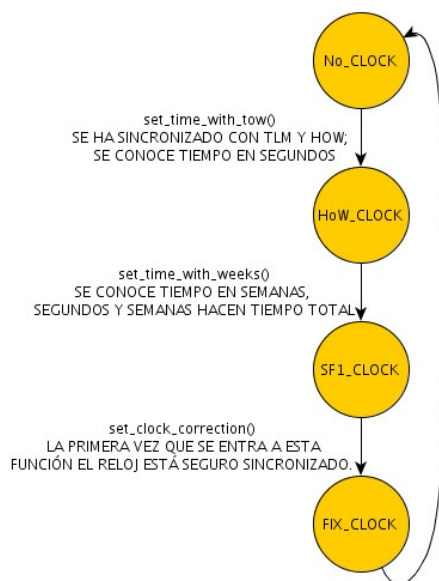


Figura 4. Relógio de controle

O programa está controlado pela execução de uma interrupção de 505 microssegundos de duração. Ele começa com a execução da função *cyg_user_start()*, onde se inicializa a máquina de estados, *Initialize_tracking()*, a estrutura de configuração dos canais dos correladores, *Initialize_allocation()*, e a interrupção que controla a execução do software, *Initialize_gp4020_interrupts()*.

Nesta interrupção se lê o registro de estado *status_b* e depois se executa o programa. Assim o programa tem duas linhas de execução:

1. Cada vez que se executa a interrupção, vai-se executar a máquina de estados que controla os correladores. É importante dizer que existe uma máquina de estados para cada um dos doze correladores.
2. A segunda linha de execução está relacionada com o registro *status_b*; o bit 13 deste registro se ativa cada 100 ms, isto é, cada vez que chega o sinal de saída dos correladores.

Estas duas linhas funcionam em paralelo, e interatuam entre elas, isto é, a linha 1 se executa a cada 505 microssegundos e faz uma varredura das doze máquinas de estado que controlam a sincronização do sinal dos satélites nos correladores; e, além disso, a cada 100ms, se executa a linha 2 que vai comprovar se os correladores estão sincronizados com os satélites e calcular a posição do receptor.

A máquina de estados que controla os correladores funciona como segue:

Channel acquisition: Vai comprovar se houve correlação no correlador com o PRN e a frequência escolhidas. A saída será maior que um valor limite (*threshold*) se houve correlação, e zero em outro caso. No caso que houve correlação, se muda ao estado *Channel confirm*; se não, se desloca a frequência de integração.

Channel confirm: Neste estado deve-se comprovar que realmente houve a correlação. Para isso, o sinal de saída dos correladores deve superar o *threshold* oito de dez vezes. Se isso acontecer, comuta-se para o estado *Channel pull_in*, se não, volta-se à *Channel acquisition* para começar de novo deslocando a frequência ou trocando o PRN (para tentar outro satélite).

Channel Pull_in: Neste estado é feito um ajuste exato da frequência do sinal do satélite com PLL digitais. Se é possível fazer este ajuste, se muda para o próximo estado, *Channel Lock*. No caso de não se poder fazer este ajuste, se muda para o estado *Channel off* para começar de novo o monitoramento com um outro PRN.

Channel Lock: Quando se chega neste estado, o sinal do satélite está estável, assim pode-se começar a decodificar o sinal de navegação. Para fazer isso, vai-se ativar a tarefa *message_thread*. Esta tarefa acumula os bits recebidos, quando ela tem uma trama completa (lembrar tramas –subframes- como na Figura 3) ativa a tarefa *ephemeris_thread*, que copia a trama completa e guarda-a. A máquina de estados não vai mudar o estado até que o sinal se desincronize. Se acontecer isso, se muda ao estado *Channel off* para atribuir outro PRN ao correlador e começar de novo.

Channel off: Este estado se produz quando o correlador não achou um satélite (o satélite não estava no horizonte). Então, vai-se atribuir um novo PRN ao correlador para procurar um outro satélite.

A linha de execução dois vai monitorizar o bit *status_b* que vai se ativar cada 100ms. Este tempo se considera suficiente para que a máquina de estados chegue até o estado *Channel lock*, isto é, estar sincronizado com o sinal do satélite. Assim cada 100ms se executa a tarefa *measure_thread*, que comprova se a máquina de estados de cada correlador está neste estado *channel_lock*; e se estiver lá, guarda a informação na estrutura *meas[channel]*, e ativa a tarefa *pseudorange_thread*. Para os canais que não chegaram até *channel_lock*, se despreza esta estrutura e não se ativa a tarefa *pseudorange_thread*.

A tarefa *pseudorange_thread* vai calcular a pseudodistância e ativar a tarefa *position_thread*.

A tarefa *position_thread* vai calcular, com quatro pseudodistâncias, a posição do receptor.

4- Mudanças no software GPL-GPS para a utilização nos satélites de órbitas baixas.

Os satélites de órbita baixa (LEO) são satélites que giram ao redor da Terra numa altura de até 2000 km (abaixo do cinturão de Van Allen) com uma velocidade de aproximadamente 28000 km/h.

O Software GPL-GPS foi designado para funcionar num receptor terrestre onde os interesses do projeto eram calcular e visualizar a posição e as efemérides dos satélites monitorados.

Os objetivos dum GPS embarcado num satélite LEO são diferentes, por isso que o software precisa ser modificado. Neste trabalho se pretende fazer uma discussão sobre as modificações que deveriam ser aplicadas ao software para otimizá-lo em relação aos objetivos do projeto.

As modificações que o software precisa podem-se dividir em duas vertentes: modificações técnicas (aquelas que precisam ser feitas para que o software funcione sem erros embarcado num satélite LEO) e modificações de otimização (aquelas que poderiam ser feitas apenas para ganhar espaço de memória, e otimizar o código). A seguir, vão-se enumerar estas modificações:

-Modificações de otimização do código:

1) LEDS de DEBUG : O software GPL-GPS tem um sistema de LEDs para conhecer o estado atual da máquina de estados para monitorar o funcionamento do DEBUG. Não se precisa de debug dentro do satélite. Assim, pode-se eliminar este módulo.

2) MÓDULO DE VISUALIZAÇÃO DE DADOS: O sistema GPL-GPS tem pronto um sistema de visualização de dados pela tela. Assim, tem configurado uma saída pela UART; e várias funções para mostrar os dados pela tela. Estas funções poderiam ser apagadas também porque não terão utilidade dentro dum satélite LEO.

-Modificações técnicas:

1) EFEITO DOPPLER: O software GPL-GPS foi desenhado para funcionar num receptor terrestre, que tem velocidade desprezível com relação à velocidade do satélite e do sinal. Neste projeto, o receptor está embarcado num satélite com velocidade de ao redor de 28000 Km/h, comparável à velocidade do satélite GPS (11000 km/h). Neste caso, precisa-se recalcular a banda de frequências que vão utilizar os correlatores.

2) THRESHOLD: O threshold é o valor mínimo que deve ter o sinal na saída do correlator para considerar que o sinal do satélite foi monitorado corretamente. Como este receptor está fora da atmosfera, o sinal vai sofrer menos atenuação. Este valor deve ser recalculado em função da órbita do satélite receptor.

3) MAPEAMENTO DAS MEMÓRIAS: Caso o projeto precise de trocar as memórias da placa Zarlink GP4020, estas estão mapeadas no arquivo gp4020.h do software. O watchdog está também neste arquivo.

4) SUBFRAMES 4 e 5 DA MENSAGEM DE NAVEGAÇÃO: No software GPL-GPS, os subframes 4 e 5 da mensagem de navegação não foram decodificados. Estes frames contém o almanaque, isto é, onde é que os outros satélites estão em cada momento do dia. Esta informação é útil quando se precisa desligar o GPS (normalmente por motivos energéticos).

5) PROTOCOLO DE COMUNICACÃO: Por último, para o módulo GPS se comunicar com a CPU do satélite e com a estação terrestre, deve-se programar um protocolo de comunicação.

5- Conclusão

O software de Clifford Kelley é uma opção possível para se designar um GPS embarcável em satélites LEO.

Para isso, deveriam ser feitas algumas mudanças no código e programar novos módulos que não existem ainda mas que são necessários neste projeto.

Este é um trabalho preliminar, as mudanças propostas apenas foram estudadas teoricamente pelo que se precisa fazer um estudo prático e os testes precisos para se certificar que estas mudanças são suficientes.

Agradecimentos

A autora quer agradecer ao Professor Marcelo Lopes de Oliveira e Souza por tê-la motivado a escrever este artigo; e aos integrantes do grupo de trabalho do SRG, (*Space Research Group*) da Universidade de Alcalá, Espanha, que tornaram possível esta pesquisa no tempo que ela esteve trabalhando lá, que a ensinaram e ajudaram no árduo caminho da designação deste projeto, entre eles os pesquisadores Agustín Martínez Hellín, Sebastián Sánchez Prieto

e os engenheiros Javier Almena, Victor A. Ruiz Escribano, Alberto Carrasco e todos os companheiros de trabalho que tornaram possível que os bons momentos estivessem acima dos difíceis.

Referências

- Alfaro, R. H. Conceptos Básicos de GPS. GeoConsutoreas, San José, Costa Rica, 2006. Disponível em <http://www.colegiotopografoscr.com/downloads2/gps-general.pdf>. Acesso em: 14 de Junho de 2012.
- Köhne, A. e Wößner, M. The GPS System/Transmitted GPS Signals. Kowona Project Home Page, Magdeburgo, Suíça. Disponível em <http://www.kowoma.de/en/gps/signals.htm>. Acesso em: 14 de Junho de 2012.
- Greenberg, A. The GPL-GPS Project Home Page. Disponível em <http://gps.psas.pdx.edu/>. Acesso em: 14 de Junho de 2012.
- Kelley, C. OpenSource GPS Home Page. Disponível em <http://home.earthlink.net/~cwkkelley/>. Acesso em: 14 de Junho de 2012.
- Dallaway, J. eCos Home Page. Disponível em: <http://ecos.sourceforge.org/>. Acesso em: 14 de Junho de 2012.
- Kelley, C.; Barnes, J.; Cheng, J. OpenSource GPS: Open Source Software for Learning about GPS. Institute of Navigation, ION GPS Conference 2002. Disponível em: <http://www.gmat.unsw.edu.au/snap/publications/opensourcегps.pdf> . Acesso em: 14 de Junho de 2012.