# PC 기반 GPS 수신기 하드웨어 모듈 및 펌웨어 개발

# Hardware and Software Implementation of a GPS Receiver Test Bed Running from PC

원필롱*, 원황휴**, 이상훈*, 박옥득*, 김현수*, 김한실***
Nguyen Phi Long*, Nguyen Hoang Hieu**, Lee SangHoon*, Park OkDeuk*, Kim HyunSu*, Kim HanSil***

**Abstract** - When developing a new GPS receiver module, the essential problems are evaluation of reliable algorithms, software debugging, and performance comparison between algorithms to find optimal solution. Most GPS receiver modules nowadays use a correlator to track signals from satellites and an MCU (Micro Controller Unit) to control operations of the entire module. The problem of software evaluation from MCU is very difficult, due to limitation of MCU resources and low ability of interfacing with user. Normally, user has to expense special tool kit for a limiting access to MCU but it is also hard to use.

This article introduces an implementation of a GPS receiver test bed using correlator GP2021 interfacing with ISA (Industry Standard Architecture) PC bus. This way can give user complete control and visibility into the operation of the receiver, then user can easily debug program and test algorithms. For this article, the least square method is implemented to test the hardware and software performance.

**Key Words** : GPS, ISA, DCO, tracking loop, PLL, FLL, least square method, navigation, algorithm.

## 1. Hardware implementation

**1.1 RF front end**: RF front end catches analog signals from satellites; down converts carrier frequency from RF 1575.42MHz analog signal to IF 1.405MHz digital signal. This part is implemented by GP2015 front end with triple conversion.

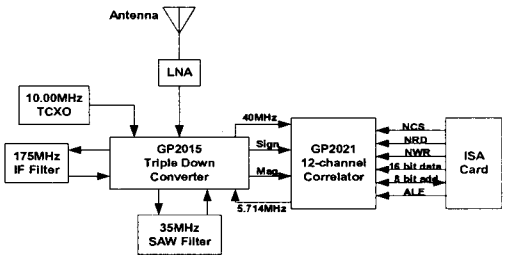


Figure 1: GPS receiver hardware components

The 10MHz TCXO is connected directly to GP2015 front end as a clock, or the "heart", for the operation of entire system. RF signal, received at 1575.42MHz, is first down converted to about 175MHz, passes through a simple band pass filter implemented by LC circuit and returns to front end. IF signal 175MHz then is down converted to 35MHz, passes through SAW filter (Surface Acoustic Wave filter), and turns back to RF front end. The performance of SAW filter is very important and strongly affects to the operations of later stages. In the final stage, conversion changes signal to about 4.3MHz. This analog signal is then sampled at clock rate of 40/7MHz or approximately 5.714MHz, provided from GP2021, to give 2-bit quantized AD converter (sign and magnitude) at frequency of approximately 1.405MHz.

**1.2 Digital signal processing**: Digital processing part converts carrier digital signal at 1.405MHz to base band, then correlates base band signal with local Gold code generator to decode navigation data transmitted from satellite. This part is implemented by GP2021 correlator.

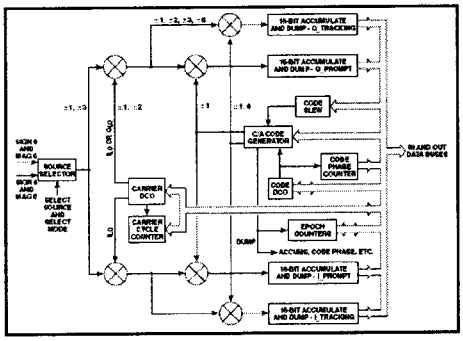

Figure 2: Tracking module inside correlator

The Carrier DCO (Digital Control Oscillator) generates replica carrier wave form and adjusted from nominal frequency of 1.405MHz to catch satellite carrier frequency.

저자 소개
* 원필롱, 이상훈, 박옥득, 김현수
: 蔚山大學校 電氣電子情報工學部 碩士課程
** 원황휴 : 蔚山大學校 自動車船泊技術大學院 碩士課程
*** 김한실 : 蔚山大學校 電氣電子情報工學部 敎授

If correct carrier is catch, this down conversion gives base band signal, which is the PRN code transmitted from satellite. In order to measure phase of carrier, both in-phase I(sine function) and quadrature Q(cosine function) are generated, and added with input signal. The difference with nominal frequency is the result of Doppler frequency and can be used to calculate receiver velocity. The Code DCO generates and adjusts frequency of C/A Code generator from nominal frequency of 1.023MHz, while the Code Slew tries slewing the replica code to match with incoming code string. The result of this correlation is dump into accumulated register. Base on this result, compensated frequency is calculated and DCOs values then will be updated.

**1.3 PC interface:** An ISA interfacing board is implemented by decoding two PC I/O ports. Port 0x300 enables the lower 8-data-bit ISA bus to send correlator register address. This port also activates signal ALE (Address Latch Enable) to GP2021. After register address has been sent and latched, 16 bits data can be read or written through port 0x302. Intel x86 interfacing timing is appropriate for latching address and transferring data.

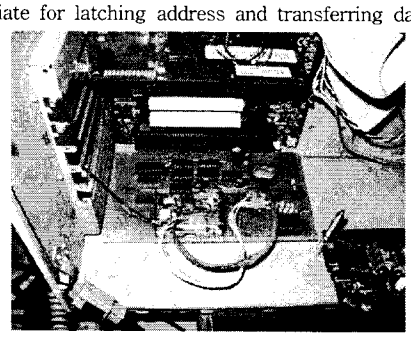

Figure 3: Real hardware connected to PC

## 2. Software implementation

Software controls the operation of GP2021, tunes channels frequency to catch and track satellites, obtains data bits and decode navigation message[2], calculates satellites positions and pseudoranges, calculates receiver position.[1]

**2.1 Tracking loops:** The software uses a single interrupt routine to handle the tracking loops and find the navigation message. All other functions are handled using a polling method triggered by flags from the interrupt routine.[3] To track the signal, one just speed up or slow down the carrier and code DCO's to keep the channel locked in code and carrier phase. The correlator read timing of the receiver is set by taking over the PC's Int 0 which is normally used to keep the computer real time clock up to date. The 8254 normally interrupts the computer about every 50ms. This software replaces this interrupt routine with its own (GPS_interrupt) and sets the interrupt time to about 500us. Whenever the interrupt occurs, channels data registers in correlator are checked to see which channels have dumped correlation data. The correlation data is read and a check is made to see if a 99.9999 ms "tic" has occurred. If it has, the measurement data is obtained and stored. Each channel is processed based on the state the channel is in.

**State 1** is the acquisition state. It does the code and frequency (Doppler) search to find a high correlation peak. If a peak in either the prompt or delay correlator (power of in-phase and quadrature signal) is above the threshold, it switches to state 2.

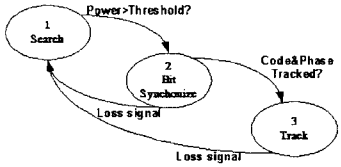


Figure 4: Receiver state chart diagram

**State 2** is the synchronizing state. First it stops the search and dwells at the code and Doppler where the high correlation peak was found in state 1 to confirm the presence of the signal, in order to reduce the false alarm rate. If signal has been confirmed, the bit synchronizing state attempts to start tracking the signal in order to pull the frequency in close enough that the carrier phase can be tracked. This state is enabled for about 1500ms, during the last 500ms of this time, the SNR (Signal to noise ratio) and phase errors are measured, and the program attempts to synchronize onto the edge of a data bit. If it confirms carrier and code tracking, it switches to state3. Since the frequency is likely to be very far off, the receiver uses a combination of a frequency locked loop FLL and a phase locked loop PLL. As the time progresses into the state the effect of the FLL is decreased so that at the end of the state the PLL is dominant.

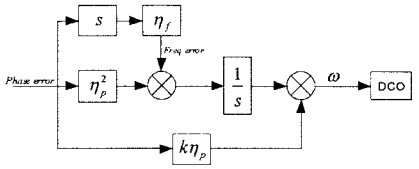


Figure 5: Second order PLL with FLL aiding

The phase error $p$ first is calculated from correlated result:

$$p(t) = atan(Q_n, I_n) - b(t)\pi \quad (1)$$

where $b(t)$=0 or 1

With sample interval $T$ = 1ms:

$$p(n) = atan(Q_n, I_n) - b(t)\pi \quad (2)$$

The frequency error :

$$f(n) = \frac{dp}{dt} = \frac{p(n) - p(n-1)}{T} \quad (3)$$

Transfer function for the loop filter:

$$\omega(s) = \left(\frac{\eta_p^2}{s} + k\eta_p\right)p(s) + \frac{\eta_f}{s}f \quad (4)$$

where $\eta_p$, $\eta_f$ are natural frequencies of PLL and FLL respectively[4].

Changing (4) to time domain, $\omega(t)$ can be calculated every loop and update for DCOs values.

**State 3** is the normal tracking state. In this state receiver has to perform the tracking loops to align data bit and phase, otherwise signal will soon be lost. Because frequency in this state does not change much, only PLL is needed. An integration over a data bit (20ms) is used to track code and 1ms to track phase. The data message is recorded and the time is synchronized to the TOW (time of week) of the data message.[2]

**2.1 Navigation calculation:** When receiver has succeeded in tracking satellites, data bits can be recorded to decode navigation message.[2] After getting enough information, software can start calculating navigation fix.[1] This paper applied least square method for navigation fix calculation.

Real range from satellite i to receiver :

$$r_i = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} \tag{5}$$

where $(x, y, z)$ : receiver position

$(x_i, y_i, z_i)$ : satellite i position

Pseudorange $p_i$ is the measurement range resulted from correlation, $p_i$ and $r_i$ has the following relationship:

$$p_i = r_i + c\delta t \tag{6}$$

where c is the speed of light, $\delta t$ is the clock offset, i.e. the difference between receiver clock and satellite clock. Since satellite clocks are precise and synchronized to each other, $\delta t$ is assumed to be the same for all satellites.

$$p_1 = r_1 + c\delta t = \sqrt{(x_1 - x)^2 + (y_1 - y)^2 + (z_1 - z)^2} + c\delta t$$
$$p_2 = r_2 + c\delta t = \sqrt{(x_2 - x)^2 + (y_2 - y)^2 + (z_2 - z)^2} + c\delta t$$
$$\vdots$$
$$p_n = r_n + c\delta t = \sqrt{(x_n - x)^2 + (y_n - y)^2 + (z_n - z)^2} + c\delta t$$

Linearization:

$$\begin{bmatrix} dp_1 \\ dp_2 \\ \vdots \\ dp_n \end{bmatrix} = \begin{bmatrix} \frac{x_1 - x}{r_1} & \frac{y_1 - y}{r_1} & \frac{z_1 - z}{r_1} & 1 \\ \frac{x_2 - x}{r_2} & \frac{y_2 - y}{r_2} & \frac{z_2 - z}{r_2} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{x_n - x}{r_n} & \frac{y_n - y}{r_n} & \frac{z_n - z}{r_n} & 1 \end{bmatrix} \begin{bmatrix} dx \\ dy \\ dz \\ d(c\delta t) \end{bmatrix} \tag{7}$$

Denote:

$$A = \begin{bmatrix} \frac{x_1 - x}{r_1} & \frac{y_1 - y}{r_1} & \frac{z_1 - z}{r_1} & 1 \\ \frac{x_2 - x}{r_2} & \frac{y_2 - y}{r_2} & \frac{z_2 - z}{r_2} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{x_n - x}{r_n} & \frac{y_n - y}{r_n} & \frac{z_n - z}{r_n} & 1 \end{bmatrix}, \Delta P = \begin{bmatrix} dp_1 \\ dp_2 \\ \vdots \\ dp_n \end{bmatrix}, \Delta X = \begin{bmatrix} dx \\ dy \\ dz \\ d(c\delta t) \end{bmatrix}$$

Then (7) can be rewrite:

$$\Delta P = A^* \Delta X \tag{8}$$

Least square equation:

$$\Delta X = (A^T A)^{-1} A^T * \Delta P \tag{9}$$

With initial $(x_0, y_0, z_0)$, receiver position then will be:

$$X(t) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_0 + dx \\ y_0 + dy \\ z_0 + dz \end{bmatrix} \tag{10}$$

## 3. Implementation Result

To check software and hardware implementation, navigation result is compared with average value, considered as the standard fix position.

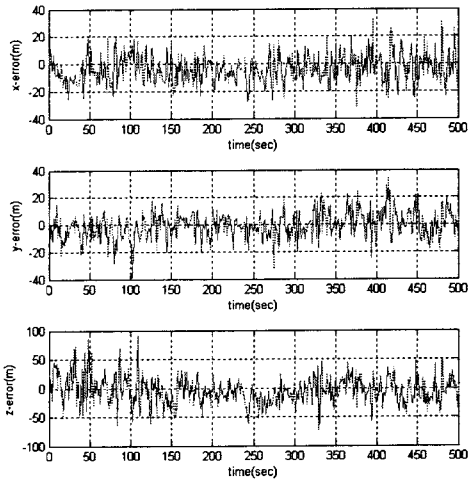


Figure 6: Experiment result

## 4. Conclusion and future work

This paper has shown that a GPS receiver can be designed and run from PC for algorithms studying and testing purpose. Although it can work, the navigation error is still large. In the future, other algorithms will be tested to find optimal solution. The code tracking loop will also be redesigned using Digital Split-Loop[5]. Noise affection from PC also needs to be considered and remove.

## REFERENCES


[1] Alfred Leick, "GPS Satellite Surveying", second edition, John Wiley & Sons, 1994.

[2] NAVSTAR GPS, "Global positioning system standard positioning service signal specification", second edition, June 2, 1995.

[3] Clifford Kelley, "Opensource GPS", http://www.home.earthlink.net/~cwkelley/

[4] W. C. Lindsey and C. M. Chie, "A survey of a digital phase-locked loops", Proc. IEEE, vol. 69, pp. 410-431, Apr. 1981

[5] J. Gustrau and Michael H. Hoffmann, "Reducing the PLL Noise Bandwidth by a Digital Split-Loop", IEEE Communications letters, vol. 3, no. 4, pp.111-112, April 1999.