Development of the low-cost RTK-GPS receiver with an open source program package RTKLIB

Tomoji Takasu [1], and Akio Yasuda [2]

Laboratory of Satellite Navigation, Tokyo University of Marine Science and Technology

[1] (Tel: +81-5245-7365, E-mail: ttaka@gpspp.sakura.ne.jp)

[2] (Tel: +81-5245-7365, E-mail: yasuda@kaiyodai.ac.jp)

Abstract: RTKLIB is an open source program package for RTK-GPS developed by the authors. RTKLIB is a compact and portable program library written in C to provide a standard platform for RTK-GPS applications. The library implements fundamental navigation functions and carrier-based relative positioning algorithms for RTK-GPS with integer ambiguity resolution by LAMBDA. RTKLIB also supports data communication via serial I/O, TCP/IP connection and NTRIP, and various data formats including RTCM 2.3, RTCM 3.1 and proprietary raw messages for some GPS receivers. By supporting RTCM and NTRIP, NRTK (Network RTK) service can be used with RTKLIB. From the version 2.2.0, RTKLIB has been distributed under the GPLv3 license. RTKLIB was originally implemented on Windows PC. In this study, we port RTKLIB to a small and compact single-board computer BeagleBoard and construct a low-cost RTK-GPS receiver with RTKLIB. BeagleBoard has 600MHz ARM Cotex-A8 core CPU and supports embedded Linux environment. In order to acquire and track GPS signals, we employ a single-frequency GPS receiver module LEA-4T provided by u-blox AG. The RTK-GPS server running on BeagleBoard inputs the u-blox raw binary data messages. The server also inputs the base-station data via a serial port or USB network device and computes RTK-GPS solution in real-time. The total cost of the developed RTK-GPS receiver was about $400. To demonstrate and verify the performance of the low-cost RTK-GPS receiver, we made some field tests. In these tests, CPU/memory usage, accuracy of solutions and fixing ratio are evaluated. According to the test results, even with such a low-cost RTK-GPS receiver, we can obtain reasonable performance in company with RTKLIB.

Keywords: RKTLIB, Open Source Software, Low-Cost RTK-GPS Receiver, BeagleBoard

1. Introduction

RTK-GPS (real-time kinematic GPS) is one of the most precise positioning technologies, with which users can obtain cm-level accuracy of the position in real-time by processing carrier-phase measurements of GPS signals. Conventionally RTK-GPS had been utilized for limited application like geodetic survey. In these days, the application of RTK-GPS has been continuously expanded to various areas like mobile mapping system, precise navigation of vehicles, construction machine control and precision agriculture. The precise positioning technology with RTK-GPS is expected to be used for much wider applications increasingly in the future.

For RTK-GPS, users usually need to prepare geodetic-grade receivers with the firmware supporting RTK-GPS or the proprietary RTK-GPS software on the receiver controller or PC provided by the receiver vendor. The receivers or such software for RTK-GPS, however, are generally still very expensive comparing to general-purpose GPS receivers. This is one of the reasons why RTK-GPS is still not popular and is used only for limited application areas. Many peoples, who require more precise position, are longing much lower cost RTK-GPS receivers.

Since several years ago, the authors have been developing a compact and portable software RTK-GPS library. We refer the library RTKLIB, which is simply derived from "RTK library" [1]. Originally RTKLIB was intended to be used for our internal research work in order to evaluate precise positioning algorithms or to provide an application platform for development of precise positioning systems. In the beginning, RTKLIB had very simple function for carrier-based relative positioning and RINEX [2] file handling for post processing. In company with several version up, a lot of useful functions and APs (application programs) for RTK-GPS were added to RTKLIB. From version 2.2.0 released in 2009, we have been distributing RTKLIB as an open source program package under the GPLv3 license [3]. The package of RTKLIB consists of user executable binary APs on Windows and whole source programs of the library and the APs. Users can freely download the program package, use the APs, install or link the library to user own AP and modify the source codes according to the requirements for user applications.

The latest version of RTKLIB supports some consumer-grade receivers which are able to output raw measurement data of GPS signals. With RTKLIB and such receivers, users can construct and operate their original low-cost RTK-GPS system. The authors have already evaluated the RTK-GPS performance with such consumer-grade single-frequency antennas and receivers by field tests [4]. Such tests were conducted in order to clarify issues to apply them to RTK-GPS. As the results of these studies, we found that the difference between consumer-grade receivers and geodetic-grade ones is not so large regarding to the receiver performance itself. With good antennas, we can obtain cm-level accuracy of the receiver position even with such low-cost receivers. However, dual-frequency receivers have an advantage of much shorter time of ambiguity resolution. With a single-frequency receiver, at least a few minutes are necessary to obtain the first fixed solution. So, in the environment with many cycle slips like for mobile vehicle navigation, low-cost receiver is not suitable for RTK-GPS. Though, for the application with continuous observation like crustal deformation monitoring, low-cost single-frequency receiver could be applicable for short baseline RTK-GPS.

In this study, we develop a RTK-GPS receiver with RTKLIB in order to demonstrate such a low-cost RTK-GPS system and clarify the issues for implementation and operation of the system. For the purpose, we also conduct some field tests to evaluate the performance of the RTK-GPS receiver. with RTKLIB

2. RTKLIB

2.1 System Requirements

RTKLIB consists of a simple program library and several application programs (APs) for RTK-GPS utilizing the library. The design goals of RTKLIB are simplicity, portability and sufficient performance. To achieve these design goals, we selected the ANSI C to write the RTKLIB library and for its APIs (application program interfaces). The RTKLIB library internally uses standard socket and pthread (POSIX thread) libraries for Linux/UNIX or winsock and WIN32 thread for Windows. For the performance improvement, RTKLIB can be built with LAPACK/BLAS or Intel MKL for fast matrix computation by setting compiler options.

The console APs in RTKLIB were also written in standard C and standard libraries. These APs can be built on many environments. We have already built them by gcc on Linux, by gcc on Mac OS X, by MS Visual Studio, Intel C or Borland C on Windows. The GUI APs in RTKLIB were written in C++ to utilize environment-dependent GUI (graphical user interface) library. Current version RTKLIB supports the GUI APs running only on Windows. These APs use Borland VCL (visual component library) for GUI functions. The distribution package of RTKLIB already includes pre-built user executable binary APs on Windows, which were built by free edition Borland Turbo C++.

2.2 Library Functions

The program library of RTKLIB provides the following various functions for positioning algorithms for RTK-GPS.

- Matrix and vector functions
- Time and string functions
- Coordinates transformation and geoid model
- Navigation processing
- Troposphere, Ionosphere and Antenna models
- Single point positioning
- Carrier-based and code-based relative positioning
- OTF (on the fly) integer ambiguity resolution
- Receiver raw binary data input
- Positioning solution/NMEA input/output
- RINEX observation data/navigation message input/output
- SP3 Precise ephemeris input
- Stream data communication library
- RTK-GPS positioning server

Current RTKLIB supports some GPS receivers' proprietary messages for raw pseudorange/carrier-phase observables and GPS satellite ephemeris. RTKLIB also implements the standard RTK-GPS messages defined by RTCM 2.3 [5] and RTCM 3.1 [6]. In addition to these messages, the stream data communication library of RTKLIB supports NTRIP (Networked Transport of RTCM via Internet Protocol) [7] for RTK-GPS communication as well as standard TCP server/client connection and serial ports. By using these RTCM and NTRIP functions provided by RTKLIB, users can utilize the network RTK service, which supports such standard formats and protocols. Refer the manual of the latest version RTKLIB [8] for library APIs and Release Notes [9] for supported receivers and messages.

2.3 Application Programs

RTKLIB provides several useful APs supporting RTK-GPS as well as carrier-based precise post processing and some utilities. The latest version of RTKLIB includes the following console and GUI APs.

- RTK-GPS positioning (RTKNAVI)
- Post-mission baseline analysis (RTKPOST, RNX2RTKP)
- Communication utility (STRSVR)
- Plot graph of solutions and observation data (RTKPLOT)
- RINEX converter of receiver raw data log (RTKCONV, CONVBIN)

To use RTK-GPS positioning AP RTKNAVI on a PC, users have to connect the PC to the GPS receivers, which output raw measurement data of both pseudorange and carrier-phase, and satellite ephemeris. Users also have to configure the input and output stream settings like serial port options, address, mount point, user ID and password for the NTRIP caster. After starting the AP, the RTK-GPS server thread receives the messages of observation data from the rover receiver and the base-station, computes solutions in real-time by RTK-GPS algorithms with them and send the solution to the output stream.

RTKLIB version 2.2.2 does not yet contain the console version of real-time positioning AP. So users are unable to use RTK-GPS without PC. In this study, we newly wrote such RTK-GPS server codes to develop the RTK-GPS receiver with RTKLIB. These codes will be included in the next release of RTKLIB.

2.4 Algorithms for RTK-GPS

In this section, we briefly introduce the positioning algorithms for RTK-GPS used in RTKLIB. For carrier-based relative positioning with a short length baseline between rover r and base-station b, the following measurement equations for carrier-phase Φ and pseudorange P in m are commonly used. In these equations, satellite and receiver clock-biases, and atmospheric effects are eliminated by double-differencing technique.

$$\begin{aligned} \Phi_{rb}^{ij} &= \rho_{rb}^{ij} + \lambda(B_{rb}^{i} - B_{rb}^{j}) + \varepsilon_{\Phi} \\ P_{rb}^{ij} &= \rho_{rb}^{ij} + \varepsilon_{P} \end{aligned} \tag{1}$$

where $()^{ij}$ and $()_{rb}$ represent single-difference between satellites and between receivers, ρ is the geometric range, λ is the carrier wave length and ε is the measurement error of these observables. $B^{i}{}_{rb}$ is single-differences of carrier-phase ambiguities in cycle. The unknown state vector $\boldsymbol{x}$ for RTK-GPS positioning is defined as:

$$\begin{aligned} \boldsymbol{x} &= (\boldsymbol{r}_r{}^{T}, \boldsymbol{B}_{L1}{}^{T}, \boldsymbol{B}_{L2}{}^{T})^{T} \\ \boldsymbol{B}_{Lj} &= (B_{rb,Lj}^{1}, B_{rb,Lj}^{2}, \ldots, B_{rb,Lj}^{m})^{T} \end{aligned} \tag{2}$$

where $\boldsymbol{r}_r$ is rover antenna position in ECEF frame. Note that RTKLIB uses single-difference instead of double-difference for the carrier-phase ambiguities to avoid the hand-over problem of reference satellites. The measurement vector $\boldsymbol{y}_k$ for the epoch t_k is defined with double-differenced carrier-phase and pseudorange measurements as:

$$\boldsymbol{y}_k = (\boldsymbol{\Phi}_{L1,}{}^{T}, \boldsymbol{\Phi}_{L2,}{}^{T}, \boldsymbol{P}_{L1}{}^{T}, \boldsymbol{P}_{L2}{}^{T})^{T} \tag{3}$$

$$\boldsymbol{\Phi}_{Lj} = (\Phi_{rb,Lj}^{12}, \Phi_{rb,Lj}^{13}, \Phi_{rb,Lj}^{14}, \ldots, \Phi_{rb,Lj}^{1m})^T$$
$$\boldsymbol{P}_{Lj} = (P_{rb,Lj}^{12}, P_{rb,Lj}^{13}, P_{rb,Lj}^{14}, \ldots, P_{rb,Lj}^{1m})^T$$

By using standard EKF (extended Kalman filter), a state vector $\boldsymbol{x}$ and its covariance matrix $\boldsymbol{P}$ can be estimated by:

$$\begin{aligned}
\hat{\boldsymbol{x}}_k(+) &= \hat{\boldsymbol{x}}_k(-) + \boldsymbol{K}_k(\boldsymbol{y}_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_k(-))) \\
\boldsymbol{P}_k(+) &= (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}(\hat{\boldsymbol{x}}_k(-)))\boldsymbol{P}_k(-) \\
\boldsymbol{K}_k &= \boldsymbol{P}_k(-)\boldsymbol{H}(\hat{\boldsymbol{x}}_k(-))(\boldsymbol{H}(\hat{\boldsymbol{x}}_k(-))\boldsymbol{P}_k(-)\boldsymbol{H}(\hat{\boldsymbol{x}}_k(-))^T + \boldsymbol{R}_k)^{-1}
\end{aligned} \tag{4}$$

where $\boldsymbol{h}(\boldsymbol{x})$, $\boldsymbol{H}(\boldsymbol{x})$ and $\boldsymbol{R}_k$ are the measurements model vector, the matrix of partial derivatives and the covariance matrix of measurement errors, respectively. These are written with the equations (1) as:

$$\boldsymbol{h}(\hat{\boldsymbol{x}}) = (\boldsymbol{h}_{\Phi,L1}{}^T, \boldsymbol{h}_{\Phi,L2}{}^T, \boldsymbol{h}_{P,L1}{}^T \boldsymbol{h}_{P,L2}{}^T)^T \tag{5}$$

$$\boldsymbol{h}_{\Phi,Lj} = \begin{pmatrix} \rho_{rb}^{12} + \lambda_{Lj}(B_{rb,Lj}^1 - B_{rb,Lj}^2) \\ \rho_{rb}^{13} + \lambda_{Lj}(B_{rb,Lj}^1 - B_{rb,Lj}^3) \\ \vdots \\ \rho_{rb}^{1m} + \lambda_{Lj}(B_{rb,Lj}^1 - B_{rb,Lj}^m) \end{pmatrix}, \boldsymbol{h}_{P,Lj} = \begin{pmatrix} \rho_{rb}^{12} \\ \rho_{rb}^{13} \\ \vdots \\ \rho_{rb}^{1m} \end{pmatrix}$$

$$\boldsymbol{H}(\hat{\boldsymbol{x}}) = \left.\frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{x=\hat{x}} = \begin{pmatrix} -\boldsymbol{DE} & \boldsymbol{0} & \lambda_{L1}\boldsymbol{D} & \boldsymbol{0} \\ -\boldsymbol{DE} & \boldsymbol{0} & \boldsymbol{0} & \lambda_{L2}\boldsymbol{D} \\ -\boldsymbol{DE} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ -\boldsymbol{DE} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{pmatrix} \tag{6}$$

$$\boldsymbol{R}_k = \begin{pmatrix} \boldsymbol{DR}_{\Phi,L1}\boldsymbol{D}^T & & & \\ & \boldsymbol{DR}_{\Phi,L2}\boldsymbol{D}^T & & \\ & & \boldsymbol{DR}_{P,L1}\boldsymbol{D}^T & \\ & & & \boldsymbol{DR}_{P,L2}\boldsymbol{D}^T \end{pmatrix} \tag{7}$$

$$\rho_r^i = \left\| \hat{\boldsymbol{r}}_r - \boldsymbol{r}^i \right\|, \rho_b^i = \left\| \boldsymbol{r}_b - \boldsymbol{r}^i \right\|, \boldsymbol{E} = (\boldsymbol{e}_r^1, \boldsymbol{e}_r^2, \ldots, \boldsymbol{e}_r^m)^T$$
$$\boldsymbol{R}_{\Phi,Lj} = 2\mathrm{diag}(\sigma_{\Phi,Lj}^1{}^2, \sigma_{\Phi,Lj}^2{}^2, \ldots, \sigma_{\Phi,Lj}^m{}^2)$$
$$\boldsymbol{R}_{P,Lj} = 2\mathrm{diag}(\sigma_{P,Lj}^1{}^2, \sigma_{P,Lj}^2{}^2, \ldots, \sigma_{P,Lj}^m{}^2)$$
$$\boldsymbol{D} = \begin{pmatrix} 1 & -1 & 0 & \cdots & 0 \\ 1 & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \cdots & -1 \end{pmatrix}$$

where $\boldsymbol{r}^i$ is satellite i position in ECEF frame, $\boldsymbol{r}_b$ is base-station antenna position, $\boldsymbol{e}^i_r$ is LOS (line-of-sight) vector from rover antenna to satellite i, and $\boldsymbol{D}$ is single-differencing matrix. For the standard deviation σ of carrier-phase or pseudorange error, RTKLIB employs a priori elevation-dependent model with user-defined parameters. The standard time update of the state vector and its covariance from epoch t_k to epoch t_{k+1} by EKF is expressed as:

$$\begin{aligned}
\hat{\boldsymbol{x}}_{k+1}(-) &= \boldsymbol{F}_k^{k+1}\hat{\boldsymbol{x}}_k(+) \\
\boldsymbol{P}_{k+1}(-) &= \boldsymbol{F}_k^{k+1}\boldsymbol{P}_k(+)\boldsymbol{F}_k^{k+1^T} + \boldsymbol{Q}_k^{k+1}
\end{aligned} \tag{8}$$

where $\boldsymbol{F}$ is the state transition matrix and $\boldsymbol{Q}$ is the covariance matrix of system noise. For the kinematic positioning mode, white noise model is usually assumed for the rover antenna position as:

$$\boldsymbol{F}_k^{k+1} = \begin{pmatrix} \boldsymbol{0}_3 & & \\ & \boldsymbol{I} & \\ & & \boldsymbol{I} \end{pmatrix}, \ \boldsymbol{Q}_k^{k+1} = \begin{pmatrix} \infty & & \\ & \boldsymbol{0} & \\ & & \boldsymbol{0} \end{pmatrix} \tag{9}$$

Considering numerical stability, RTKLIB resets the states of the rover antenna position to the single point solution at every epochs instead of the pure kinematic model expressed by (9). In this scheme, the iteration of the filter due to the nonlinearity of the measurement equations also can be avoided for efficient computation. In the static positioning mode, RTKLIB uses just a simple state transition model defined as $\boldsymbol{F} = \boldsymbol{I}$ and $\boldsymbol{Q} = \boldsymbol{0}$. Current version only supports the kinematic or the static mode, which does not incorporate any receiver dynamics. Regarding to the single-differenced carrier-phase ambiguity term, the initial state is determined as guess estimated value with single-differenced carrier-phase minus pseudorange measurement. If a cycle-slip detected, the state of the carrier-phase ambiguity is reset to initial value in the same manner. To detect the cycle-slips, RTKLIB monitors the jump of the geometry-free LC (linear combination) of L1 and L2 carrier-phase as well as LLI (loss of lock indicator) or lock-time provided by the receiver.

By solving the EKF formula (4) with the RTK-GPS equations described above, the estimated rover antenna position and carrier-phase ambiguities can be obtained. The estimated rover antenna position is frequently referred as "FLOAT" solution without integer ambiguity resolution.

2.5 Integer Ambiguity Resolution

Once estimated states obtained, the float carrier-phase ambiguities should be resolved into integer values in order to improve accuracy and convergence time. In RTKLIB, the float solution and the covariance matrix are transformed to double-differenced forms by:

$$\begin{aligned}
\hat{\boldsymbol{x}}'_k &= \boldsymbol{G}\hat{\boldsymbol{x}}_k(+) = (\hat{\boldsymbol{r}}_r{}^T, \hat{\boldsymbol{N}}^T)^T \\
\boldsymbol{P}'_k &= \boldsymbol{G}\boldsymbol{P}_k(+)\boldsymbol{G}^T = \begin{pmatrix} \boldsymbol{Q}_R & \boldsymbol{Q}_{NR} \\ \boldsymbol{Q}_{RN} & \boldsymbol{Q}_N \end{pmatrix} \\
\boldsymbol{G} &= \begin{pmatrix} \boldsymbol{I}_3 & & \\ & \boldsymbol{D} & \\ & & \boldsymbol{D} \end{pmatrix}
\end{aligned} \tag{10}$$

where $\boldsymbol{N}$ is the double-differenced carrier-phase ambiguities, which should be integers by canceling the receiver initial phase terms. In this form, the best integer vector $\breve{\boldsymbol{N}}$ is searched to satisfy the condition of ILS (integer least square) problem as:

$$\breve{\boldsymbol{N}} = \underset{\boldsymbol{N} \in \boldsymbol{Z}}{\mathrm{argmin}}((\boldsymbol{N} - \hat{\boldsymbol{N}})^T \boldsymbol{Q}_N{}^{-1}(\boldsymbol{N} - \hat{\boldsymbol{N}})) \tag{11}$$

To solve the problem, a well-known efficient strategy LAMBDA [10] and its extension MLAMBDA [11] are employed for RTKLIB. After the validation by the simple ratio-test, "FIX" solution of the rover antenna position is obtained by solving the following equation.

$$\breve{\boldsymbol{r}}_r = \hat{\boldsymbol{r}} - \boldsymbol{Q}_{RN}\boldsymbol{Q}_N{}^{-1}(\hat{\boldsymbol{N}} - \breve{\boldsymbol{N}}) \tag{12}$$

3. Low-Cost RTK-GPS Receiver with RTKLIB

3.1 Selection of GPS Receiver Board/Module

Table 1 shows the GPS receiver boards or modules supporting raw measurement and satellite ephemeris output with less than $300 as the sample price. All of the receivers in the table are only for single-frequency signals. Currently there is no dual or triple frequency receiver available in reasonable price range. Out of these receivers listed in Table 1, we select LEA-4T provided u-blox AG [12]. The reasons why we select the receiver are:

- Good performance was obtained by the previous test [4].
- It is a small and compact module with low-power consumption.
- It provides "half-cycle ambiguity resolved" carrier-phase.
- It supports frequent update rate of raw data up to 10 Hz.

LEA-4T can be configured to output raw measurement data as RXM-RAW messages and navigation data frame buffers as RXM-SFRMB according to the UBX binary protocol [13]. LEA-4T also supports an asynchronous serial port and a USB interface to communicate with the host CPU.

Table 1. GPS Receiver Boards/Modules Supporting Raw Measurement and Satellite Ephemeris Output

Vender	Receiver Board/Module	B/M *1	# of CH	Max Raw Rate	Sample Price
NovAtel	SuperStarII	B	12ch	1Hz	$165
NovAtel	OEMStar*2	B	14ch	10Hz	?*4
Magellan	AC12	M	12ch	1Hz	$106
SiRF	SiRFstarII	C	12ch	1Hz	$57*5
GARMIN	GPS 15L/15H	M	12ch	1Hz	$60
u-blox	LEA-4T	M	16ch	10Hz	$179
u-blox	LEA-5T*3	M	50ch	2Hz	$179
u-blox	LEA-6T	M	50ch	?	?*6
Hemisphere	Crescent	B	12ch	10Hz	$285
SkyTraq	S1315F	M	12ch	20Hz	$25

*1 B: OEM Board, M: Module, C: Chip, *2 supports GLONASS, *3 F/W 6.00, *4 2009/4Q, *5 Module, *6 2010/1Q

3.2 Hardware Configuration

Figure 1 shows the hardware configuration of the developed low-cost RTK-GPS receiver. To construct the receiver, we port RTKLIB to a very compact and small single-board computer BeagleBoard [14]. Table 2 shows the features of BeagleBoard. BeagleBoard has 600MHz ARM Cortex-A8 core CPU and supports embedded Linux. For the receiver, we develop a additional small receiver board. On the receiver board, a LEA-4T modules is mounted, which connects to the a UART port of the expansion connecter of BeagleBoard via a voltage level converter. On the receiver board, an optional 6-axis ADI 165xx MEMS IMU also can be mounted by using SPI port of the extension connector of BeagleBoard. The receiver module LEA-4T has a SMA connector for an external GPS antenna. On BeagleBoard, user can connect some peripherals like WiFi LAN card and HSDPA (high speed downlink packet access) modem via two USB ports to communicate with external receivers or the base stations. BeagleBoard also supports SD card to save operational log or raw data as the RTK-GPS receiver. Table 2 shows a picture of the developed RTK-GPS receiver.

Table 2. Features of BeagleBoard (Rev C)

Item	Feature
Processor	TI OMAP3530 - ARM Cortex-A8 core CPU 600Hz - TMS320C64x+ DSP 430MHz
Memory	256MB SDRAM+256MB NAND Flash
Ext. Memory	MMC+/SD/SDIO
Serial I/F	RS-232C
USB I/F	USB 2.0 EHCI HS +USB 2.0 HS OTG
Peripherals	DVI-D, S-Video, Audio I/O, LCD I/F (Rev C)
Expansion	28P Header Pin (I2C, I2S, SPI, MMC/SD, UART
Power Supply	5V, about 350mA
Board Size	3" × 3" (76.2 × 76.2 mm)
Price	$149 (Digi-Key)

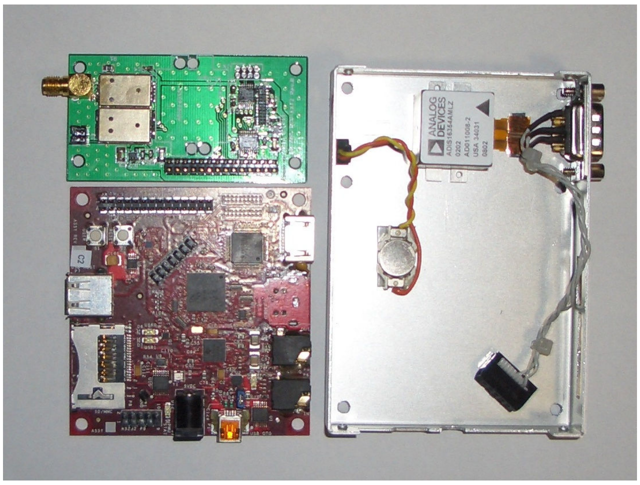

Figure 2. Picture of the developed Low-Cost RTK-GPS Receiver. Upper left shows the receiver board, lower left is BeagleBoard (Rev C), and right is the case, cables, connector and optional MEMS IMU device.

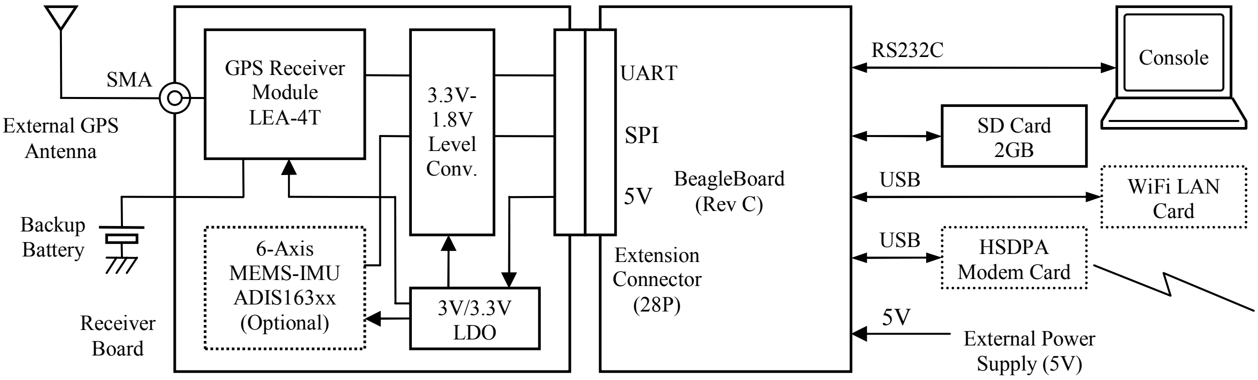

Figure 1. Hardware Configuration of Low-Cost RTK Receiver with RTKLIB

Table 3 shows the parts and price list of the receiver. Without the optional parts, the total cost for a set of the developed RTK-GPS receiver is approximate $400. Note that this is not include the cost of the external GPS antenna and the cable, USB peripherals for communication and the power supply.

Table 3. Parts and Prices of the RTK-GPS Receiver

No	Parts	Specs	Provider	#	Price
1	BeagleBoard Rev.C2	OMAP3530, 256+256MB RAM/Flash	Digi-key	1	$149
2	LEA-4T	16ch, Single-Freq GPS Receiver Module	u-blox	1	$179
3	Extension Board	3" x 1.2", double-side	Silver Circuit	1	$18
4	TPS79933DDCR	IC LDO Reg 200mA 3.3V TSOT-23-5	TI	1	$1
5	TPS79930DDCT	IC LDO Reg 200mA 3.0V TSOT-23-5	TI	1	$1
6	TXS0108E	IC 8bit Non-Inv Transtr 20TSSOP	TI	1	$2
7	TXS0104E	IC 4bit Non-Inv Transtr 14TSSOP	TI	1	$2
8	Connectors	SMA, D-Sub-9P, Header-28P-M/F	-	1s	$20
9	Chip Cap, Reg.	-	-	1s	$2
10	Case YM-115	115 x 80 x 20 mm	Takachi	1	$6
11	Screws, Spacers	-	-	1s	$3
12	SD Card	2GB	-	1	$20
	Total	-	-	-	$403
OP1	ADIS16354	6-Axis MEMS-IMU, 1.7g, 300deg/s, SPI	ADI	1	$720
OP2	CLM-112-02	24P 1mm-pitch sockets	Samtech	1	$7

3.2 Software Configuration

Table 4 summarizes the software environment to port RTKLIB to BeagleBoard. For the low-cost RTK-GPS receiver, we use Ubuntu 9.04 ARM port with Linux kernel 2.6.29 with the patch. The package includes standard C library supporting the floating point coprocessor VFP in OMAP3530 CPU for efficient computation. After completing the Linux kernel build on the host environment and the generated kernel image (uImage) is saved to the SD card to boot Linux kernel from the SD card.

Table 4. Software Configuration to Port RTKLIB to BeagleBoard

Item	Description
Kernel	Linux 2.6.29-OMAP1+Patchs
Cross Compiler	ARM-gcc 4.2.1
LIBC	glibc 2.9, libc6-vfp
Root File System	Ubuntu 9.04 for ARM
Boot Loader	U-boot 1.3.3
RTKLIB	version. 2.3.0b
Host Environment	Ubuntu 9.04 + VMWare Workstation 6.5.3 on Windows Vista Home Premium 64bit

The version of RTKLIB to be ported is 2.3.0b the CPU board. From this version, command line real-time positioning AP RTKRCV has been included in the package. RTKRCV creates a RTK-GPS server thread, which acquires raw measurement data or satellite ephemerides from rover and base-station receivers, compute the positioning solutions, resolve integer ambiguities and output the solutions in real-time. RTKRCV also support the control and monitor console via the standard I/O or TELNET login from a remote terminal.

4. Evaluation of RTK Receiver with RTKLIB

4.1 Settings of Field Test

Figure 2 shows the Test configuration to evaluate the developed low-cost RTK-GPS receiver with RTKLIB. A dual-frequency GPS antenna NovAtel GPS-702-GG was mounted on the roof-top under good sky view connecting to the low-cost RTK-GPS receiver with RTKLIB. E-Mobile H21HW USB modem card was also connected to the receiver via a USB-hub. E-Mobile provides mobile internet access service with HSPDA in Japan area up to 7.2 Mbps as the download data rate. Via the mobile Internet access, the receiver connects to the NGS (Nippon GPS data service) NTRIP caster providing 1 Hz measurement data of the GEONET stations. For the field test, we utilized 0979 Yamanashi-Takane station nearest to the RTK-GPS receiver where the baseline length was 6.1 km. The GPS measurement data and the antenna position were transmitted as RTCM 3.1 message 1004 and 1005 through the NTRIP connection. The update rate of the RTK-GPS solution was set to 10 Hz by configuring the LEA-4T module raw measurement output rate. The solution and path-through log of the rover and the base-station inputs were configured to be recorded to the SD card.

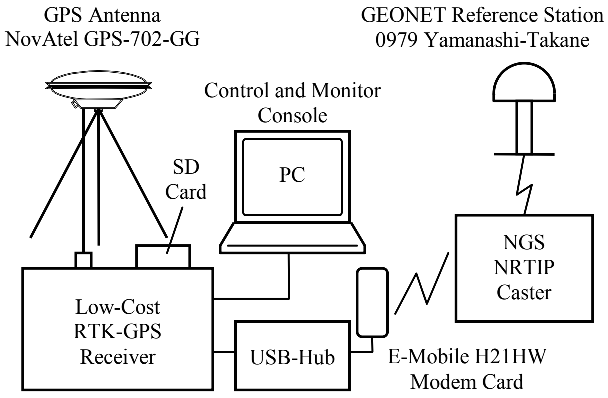

Figure 2. Test Configuration of Low-Cost RTK-GPS Receiver

4.2 CPU/Memory usages and Stream I/O bit-rates

By using Linux TOP command, we measured CPU and memory usage of the RTK-GPS receiver during the real RTK-GPS process running. The stream I/O bit-rates are also measured by RTKRCV stream monitor command. Table 5 shows measured CPU/memory usage and Stream I/O bit-rage. The most of CPU and Memory usage is due to the RTKRCV AP.

Table 5. CPU/Memory Usage and Stream I/O Bit-Rate

Item	Sub-Item	Results	Unit	Notes
CPU Usage	User	18.2 - 20.1	%	
	System	1.0 - 2.6	%	
Memory Usage	Total	239616	KB	
	Used	53996	KB	
	Free	185320	KB	
Stream I/O Bit-Rate	Rover Input	27.3 - 31.0	kbps	Serial
	Base Input	1.1 - 1.4	kbps	NTRIP
	Sol Output	11.2 - 11.3	kbps	File
	Rover Log	26.1 - 31.1	kbps	File
	Base Log	1.1 - 1.3	kbps	File

According to the results, 20 Hz update of RTK-GPS solution with integer ambiguity resolution seems practical with BeagleBoard and RTKLIB. Some other applications like machine control also can be executed simultaneously using the remains of CPU time and memory usage of BeagleBoard.

4.3 Accuracy and Fixing Ratio

Under the receiver configuration describe above, we recorded the solution output log of the low-cost RTK-GPS receiver to the SD card. The log started at 9:31 GPST and stop at 11:39 on September 30 2009. Total 76971 kinematic solutions were obtained for approximately 2 hours. Figure 3 shows the satellite geometry during the test. The average GDOP was 3.1 and the number of visible satellite was from 5 to 8 at the elevation mask angle of 15°. After the test we compared the log to reference position and evaluated the performance of the developed low-cost RTK-GPS receiver.

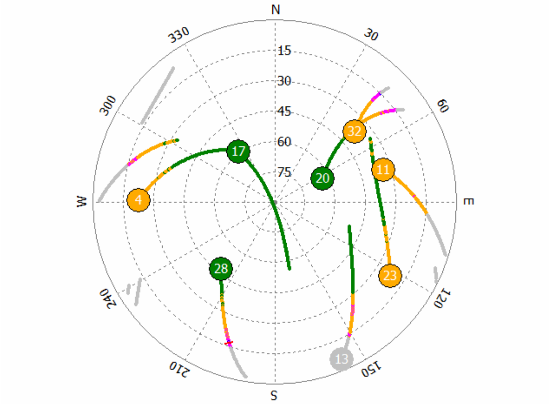

Figure 3. Visible Satellites for the Receiver Test

Figure 4 shows the E-W, N-S and U-D components of the position errors, which are referenced to the precise static solution obtained by using a dual-frequency geodetic grade receiver. In the figure green dots indicates the fixed solutions and the oranges are float solutions. Table 6 summarizes the fixing ratio and RMS errors of these solutions.

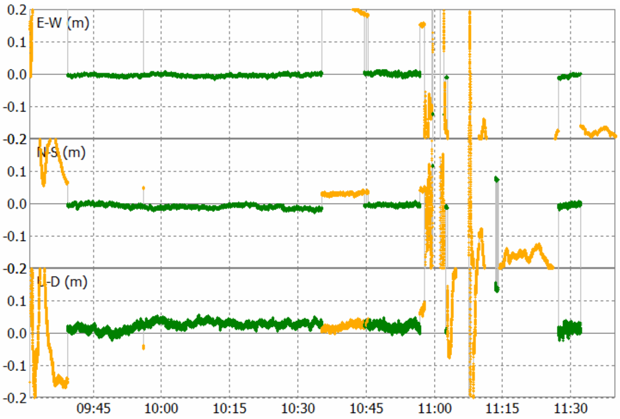

Figure 4. Position Error of RTK-GPS Solutions.

Table 6. Fixing Ratio and RMS error of RTK-GPS Solutions, 2009/9/30 9:31 - 11:39 GPST (76971 Epochs)

	Fixing Ratio	RMS Errors of Fixed Solutions		
		E-W	N-S	U-D
RTK-GPS Solutions	56.9%	3.0 cm	4.9 cm	7.6 cm

In Figure 4, a few miss-fixed solutions can be clearly seen at 11:00-11:15 time frame. These miss-fix solutions seems to degrade the total accuracy. Without these solutions, the receiver achieves standard RTK-GPS accuracy 1 cm + 1 ppm × baseline length as the horizontal RMS error. As the fixing ratio 50- 60 % value is considered to be reasonable performance as the single frequency receiver at the baseline length of 7 km. Generally speaking, the developed low-cost RTK-GPS receiver with RTKLIB has comparable performance to single-frequency geodetic-grade RTK-GPS receivers.

5. Conclusions

In this paper, we introduced the functions, the application programs and the algorithms for RTKLIB, which is an open source program package for RTK-GPS. We also describe the detailed design of the low-cost RTK-GPS receiver with RTKLIB. To demonstrate and verify the performance of the developed low-cost RTK-GPS receiver, we made some field tests. In these tests, CPU/memory usage, accuracy of solutions and fixing ratio are evaluated. According to the test results, even with such a low-cost RTK-GPS receiver, we can obtain reasonable performance in company with RTKLIB.

Reference

1. Takasu, T., Kubo, N. and Yasuda, A., Development, evaluation and application of RTKLIB: A program library for RTK-GPS, GPS/GNSS symposium 2007, Tokyo, Japan, 20-22 November (2007) (in Japanese)
2. Gurtner, W., RINEX: The Receiver Independent Exchange Format Version 2, 1997
3. RTKLIB: An Open Source Program Package for RTK-GPS, *http://gpspp.sakura.ne.jp/rtklib/rtklib.htm*
4. Takasu, T., and Yasuda, Evaluation of RTK-GPS Performance with Low-cost Single-frequency GPS Receivers, International Symposium on GPS/GNSS 2008, November 11-14, Tokyo International Exchange Center, Japan (2008)
5. RTCM Recommended Standards for Differential GNSS (Global Navigation Satellite Systems) Service version 2.3, August 20 (2001)
6. RTCM Standard 10403.1 for Differential GNSS (Global Navigation Satellite Systems) Services - Version 3, October 27 (2006)
7. RTCM Recommended Standards for Networked Transport of RTCM via Internet Protocol (Ntrip) version 1.0, September 30 (2004)
8. RTKLIB ver. 2.2.2 Manual, September 7 (2009)
9. RTKLIB ver. 2.2.2 Release Notes, September 7 (2009)
10. Teunissen, P. J. G., The least-square ambiguity decorrelation adjustment: a method for fast GPS ambiguity estimation, *J. Geodesy,* vol.70 (1995)
11. Chang, X.-W., Yang, X. and Zhou, T., MLAMBDA: A modified LAMBDA method for integer least-squares estimation, *J. Geodesy*, vol.79 (2005)
12. u-blox, LEA-4T ANTARIS(R) 4 Programmable GPS Module with Precision Timing Data Sheet, May 11 (2006)
13. u-blox, ANTARIS Positioning Engine Protocol Specification, Version 5.00 (2003)
14. beagleboard.org, BeagleBoard System Reference Manual Rev C3, May 6 (2009)