GPS

AssistNow Online

u-blox A-GPS Service: Implementation for u-blox GPS Receivers

Application Note

locate, communicate, accelerate

Abstract

This document explains how u-blox customers can implement access to u-blox' free AssistNow Online A-GPS service for assisting u-blox GPS receivers.

Document Information	
Title	AssistNow Online
Subtitle	u-blox A-GPS Service: Implementation for u-blox GPS Receivers
Document type	Application Note
Document number	GPS-SW-09008-C1
Document status	Released This document replaces GPS.G4-SW-05017-D and GPS-G4-SW-05022

Document status information	
Objective Specification	This document contains target values. Revised and supplementary data will be published later.
Advance Information	This document contains data based on early testing. Revised and supplementary data will be published later.
Preliminary	This document contains data from product verification. Revised and supplementary data may be published later.
Released	This document contains the final product specification.

Contents

1 Introduction

u-blox maintains a world-wide GPS monitoring network which collects ephemeris and almanac data from the GPS satellites. This data is aggregated and used to derive aiding data which is then made available via the internet, from u-blox' AssistNow Online Server.

This aiding data allows u-blox GPS receivers to have a faster Time To First Fix (TTFF) in situations where the receiver has difficulty collecting required data, and where Internet connectivity is available.

AssistNow Online is u-blox' free end-to-end A-GPS service that allows authorized users to connect to the u-blox AssistNow Online Server. AssistNow Online is an easy-to-integrate A-GPS solution that provides u-blox GPS receivers with GPS aiding data enabling a rapid TTFF, even in adverse conditions.

This document describes the protocol used between the server and clients and provides a reference implementation for a simple client.

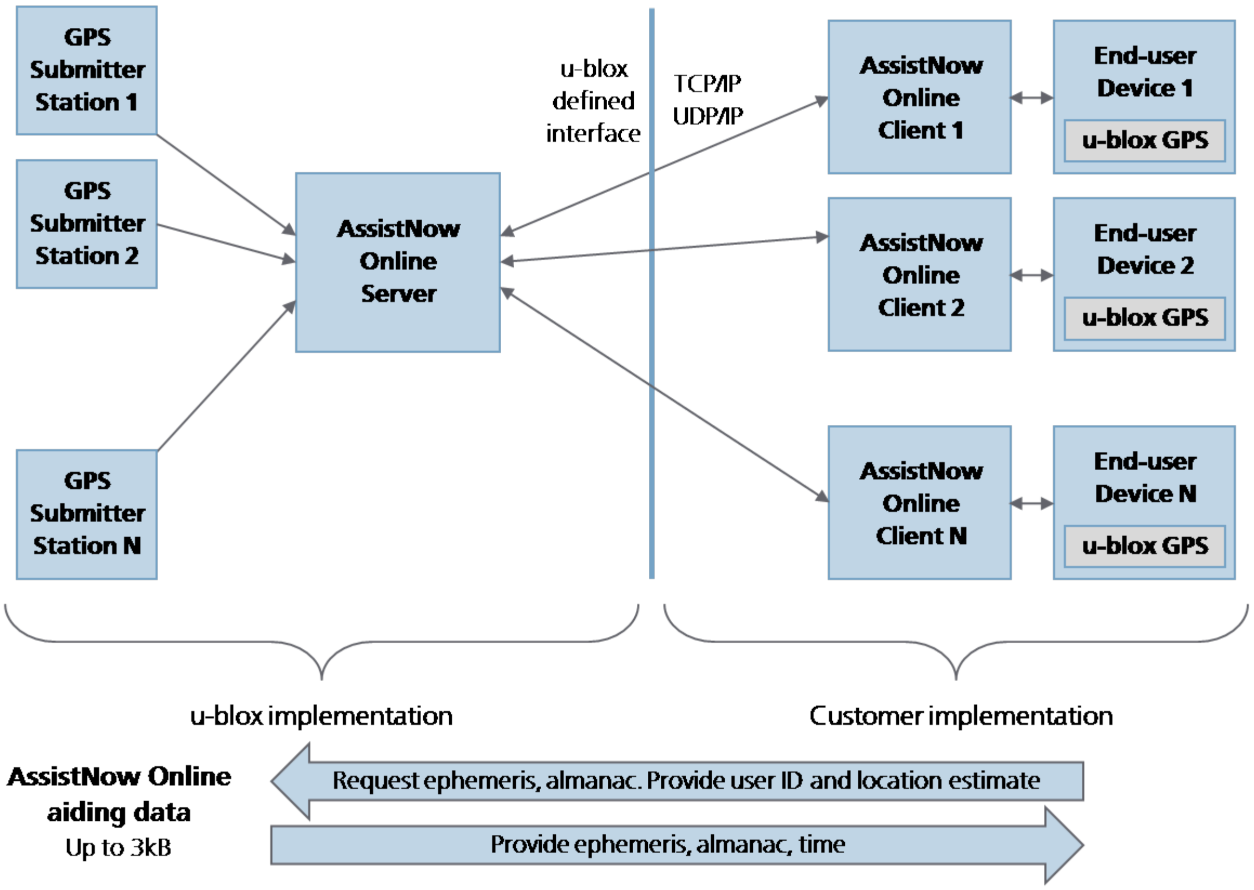


Figure 1: System Overview of AssistNow Online

2 Accessing AssistNow Online

2.1 Registering for an AssistNow Online A-GPS account

To register for AssistNow Online, simply send an e-mail to agps-account@u-blox.com. An automatic reply will be sent back with the following information:

- User name (same as e-mail address)
- Password
- Disclaimer / Terms of Use

The account will be activated within 60 minutes or less. Please read the disclaimer and terms of use carefully before proceeding.

2.2 Evaluating AssistNow Online

Two versions of u-blox' GPS evaluation tool u-center are available on our website: u-center for PCs and u-center Mobile for PDAs. u-center can be used for testing and evaluation of AssistNow Online. u-center can submit A-GPS requests and receive aiding data from the u-blox AssistNow Server, transfer this data directly into the GPS receiver and initiate aided startups.

The AssistNow Online functionality can only be used if u-center has internet access. The AssistNow Online popup window is accessed either via the u-center *Receiver* -> *Action* menu or the symbol of a thermometer with a red cross in the "Action Toolbar".

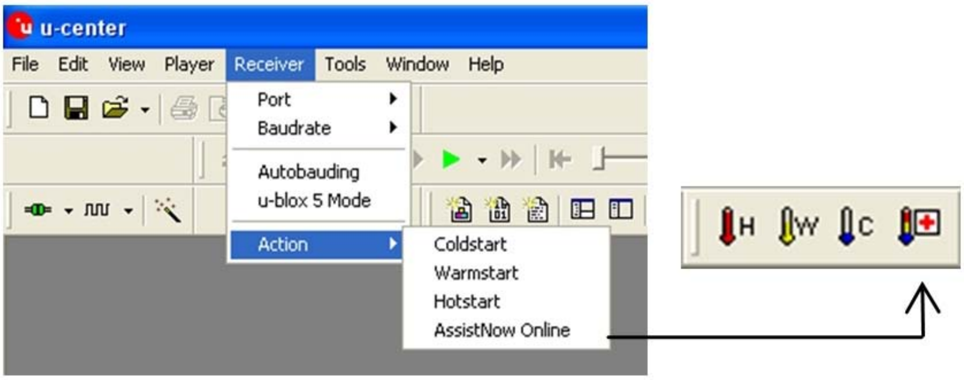
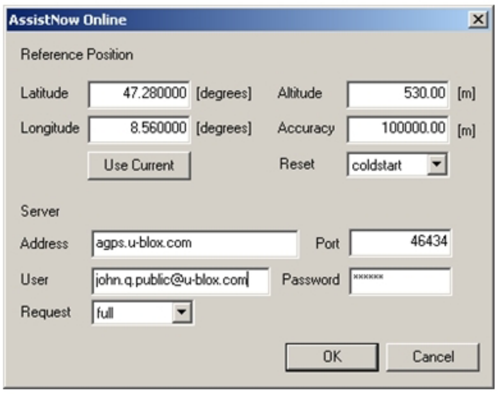


Figure 2: Accessing AssistNow Online in u-center

To use the AssistNow Online functionality in u-center:

1. Connect an ANTARIS 4, u-blox 5 or u-blox 6 based GPS receiver to u-center and make sure that the communication between u-center and the GPS receiver is working (i.e. messages from the GPS receiver are visualized in the usual way).
2. Click on the "AssistNow Online" button. The "AssistNow Online" window appears.
3. When using AssistNow Online for the first time, specify the user name (typically e-mail address) and password. User name and password will remembered for later use.
4. The TTFF performance depends on the distance to the last known position (= Reference Position) and the accuracy figure. Specify the last known position or estimated position and accuracy estimation.
5. Click on OK to start the download.



AssistNow Online

Reference Position

Latitude 47.280000 [degrees] Altitude 530.00 [m]

Longitude 8.560000 [degrees] Accuracy 100000.00 [m]

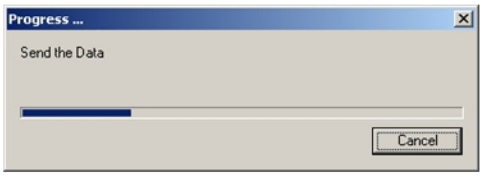
Reset coldstart

Server

Address agps.u-blox.com Port 46434

User john.q.public@u-blox.com Password ******

Request full



Progress ...

Send the Data

Figure 3: AssistNow Online window **Download in progress**

6. The GPS receiver will be restarted and downloaded assistance data will be forwarded to the GPS receiver immediately.
7. On u-center, observe the acquisition performance by viewing satellite signal levels and the "Data" docking window. This docking window displays the actual TTFF after a first position fix has been made.

☞ The "Accuracy" figure must contain a realistic value. For example, if the distance from the last known position is less than 30 Km, then specify 30000.00 m. Entering an unreasonably high accuracy figure (e.g. 9’999’999.99), or wrong reference position coordinates (e.g. Latitude = 0°, Longitude = 0°) will not provide the benefits of faster TTFF, and may even degrade performance

2.3 Accessing AssistNow Online aiding data

The AssistNow Online system is designed so that end-user devices (containing u-blox GPS receivers) can access aiding data such as Ephemeris, Almanac and Time from the AssistNow Online Server (See Figure 1). The AssistNow Online Client submits a request to the AssistNow Online Server. The AssistNow Online Server responds by providing the client with ephemeris and almanac data, which is then transferred to the u-blox GPS receiver in the end-user device. Note that the AssistNow Online Client software may be embedded in the end-user device, or a separate application, as in the case of u-center.

3 Overview

3.1 Setup

The following diagram shows the overall system setup. This document describes the functions to be implemented in {4}, and the protocol {3} to be used for communication between the AssistNow Online Server and AssistNow Online Clients.

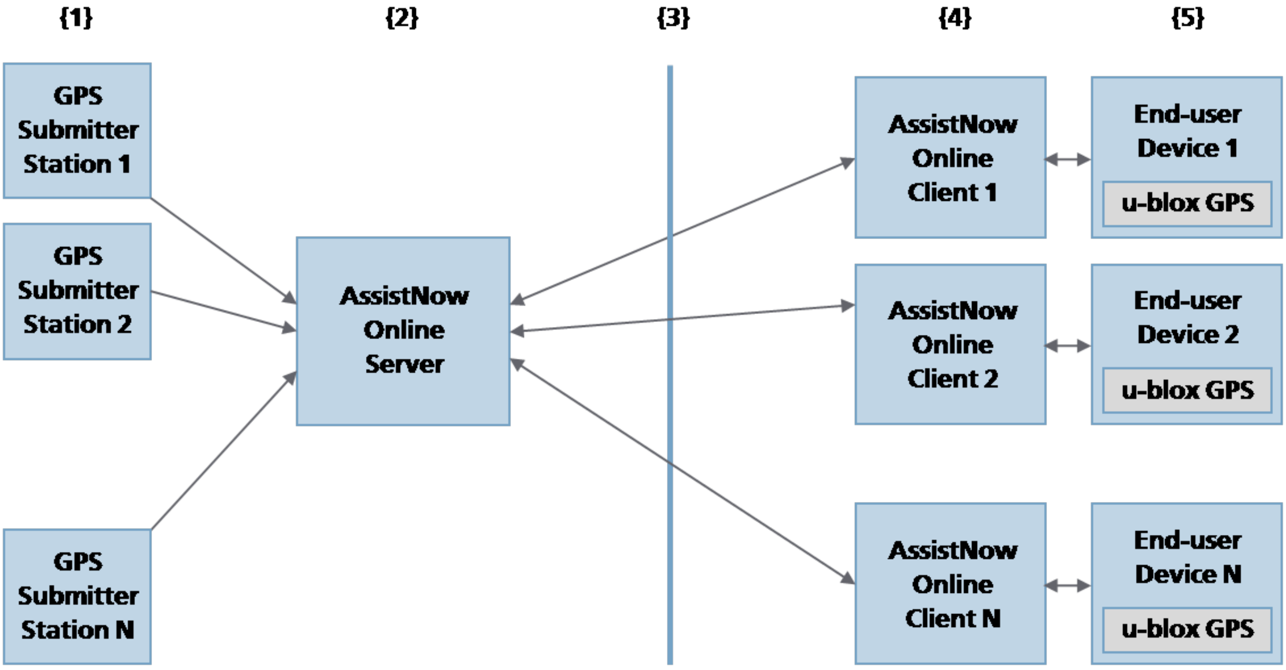


Figure 4 Assist Now Online Components

3.2 GPS Submitter Station function

GPS Submitter Stations {1} are reference GPS receivers located worldwide. These stations monitor the GPS constellation continuously and submit their measurements to the AssistNow Online Server.

3.3 AssistNow Online Server function

The AssistNow Online Server {2} collects ephemeris and almanac data from the submitter stations. This information is aggregated and used to derive aiding data which is made available to the AssistNow Online Clients via the internet.

The server listens on specific TCP/IP and UDP/IP ports to which the clients submit requests. The server answers these requests, returning whatever information the client requested and the server has available.

The server requires that clients authenticate using a username/password combination. It also requires that each client provides an approximate position of its own location, since the information sent by the server is localized to the client’s location, in order to save bandwidth.

3.4 AssistNow Online Client function

The AssistNow Online Clients' {4} function is to connect to the server, request the information, and evaluate the results. In case of a successful response, the client should then transmit the information received (without the status headers) to a u-blox receiver in the end-user device {5} through its physical connection (USB, UART, SPI, DDC), possibly relayed through some networks.

3.5 Supported receivers

The following u-blox receivers are able to process the information delivered by the server

- All u-blox 6 receivers
- All u-blox 5 receivers
- All ANTARIS 4 receivers, running firmware versions 4.00 or 4.10
- All ANTARIS receivers, running firmware versions 3.01, 3.04 or 3.10

4 Detailed description of the protocol

This section describes the protocol {3} between the u-blox AssistNow Online Server, and the AssistNow Online Client requesting information.

The information exchange is loosely built around the HTTP protocol. Upon reception of a URL-like request, the server will respond with an HTTP-style header and content. After delivery of all data, the server will terminate the connection.

4.1 Request format

The request is sent from the client to the server, immediately after establishing the socket connection.

A request is sent in plain ASCII, and has the following form

```
key=value;key=value;key=value;…\n
```

The following rules apply:

- The order of keys is not important.
- Keys and values are case sensitive.
- Key/value pairs must be separated by semicolons
- The request must be terminated with a newline character (i.e. not literal backslash-n, but the newline character, i.e. the single byte hex value 0A)

The following keys are supported

<table>
<tr><th>Key Name</th><th>Unit/Range</th><th>Mandatory/Optional</th><th>Comment</th></tr>
<tr><td>Cmd</td><td>String</td><td>Mandatory</td><td>This key determines what kind of information the client requests from the server.
<table>
<tr><th>cmd</th><th>Position</th><th>Time</th><th>Ephemeris</th><th>Almanac</th></tr>
<tr><td>full</td><td>●</td><td>●</td><td>●(*)</td><td>●</td></tr>
<tr><td>aid</td><td>●</td><td>●</td><td>●(*)</td><td></td></tr>
<tr><td>eph</td><td></td><td></td><td>●(*)</td><td></td></tr>
<tr><td>ephfull</td><td></td><td></td><td>●</td><td></td></tr>
<tr><td>alm</td><td></td><td></td><td></td><td>●</td></tr>
</table>
(*) Please note that in these cases, ephemeris for currently visible satellites at the requested position is returned</td></tr>
<tr><td>User</td><td>String</td><td>Mandatory</td><td>The username, for example "foo@bar.com". This must be a valid Email Address, as important server maintenance messages will be sent to this address</td></tr>
<tr><td>Pwd</td><td>String</td><td>Mandatory</td><td>The password. This field is transmitted in clear text, so it shall not be a safety critical password. Passwords are assigned by u-blox and are only valid for a given Email address.</td></tr>
</table>

Key Name	Unit/Range	Mandatory/Optional	Comment
lat lon	Numeric [degrees]	“Half”-Mandatory Either lat/lon or ex/ey/ez is mandatory	Approximate user position in WGS-84 Latitude / Longitude in units of degrees and fractional degrees. Example: ”lat=47.2;lon=8.55”.
Alt	Numeric [meters]	Optional	Approximate user altitude above WGS84 Ellipsoid in units of meters.
ex ey ez	Numeric [meters]	“Half”-Mandatory Either lat/lon or ex/ey/ez is mandatory	Approximate user position in ECEF Frame in units of meters Example: ”ex=4286464.77;ey=645609.71;ez=4663731”.
Pacc	Numeric [meters]	Optional	Approximate accuracy of submitted position (either lat/lon or ex/ey/ez). If this value is not provided, the server assumes an accuracy of 300km.
Latency	Numeric [seconds]	Optional	Typical latency between the time the server receives the request, and the time when the assistance data arrives at the GPS receiver. The server uses this value to correct the time being transmitted to the client (if cmd=full or cmd=aid). Example: ”latency=0.27”.

Example:

```
cmd=aid;user=foo@bar.com;pwd=whatever;lat=47.28;lon=8.56;pacc=1000\n
```

4.1.1 Requirements for the position and time parameters

The position that is being sent to the server is used for two purposes:

- The server determines the currently visible satellites at the user position, and only sends ephemeris data of those satellites which should be in view at the location of the user. This reduces bandwidth requirements. This is true for the commands aid, eph and full. Also note that the ‘pacc’ value is taken into account, meaning that the server will return all SVs visible in the given uncertainty region
- The server also feeds back the position and optional accuracy to the user’s u-blox receiver. Depending on the accuracy of the provided data, the receiver can then choose to select a better startup strategy. For example, if the position is accurate to 100km or better, the u-blox receiver will choose to go for a more optimistic startup strategy. This will result in quicker startup time. The receiver will decide which strategy to choose, depending on the ‘pacc’ parameter. If the submitted user position is less accurate than what is being specified with the ‘pacc’ parameter, then the user will experience prolonged or even failed startups.

The time that is being sent to the receiver is the time when the request arrived at the server, corrected by the optional ‘latency’ value. The client needs to ensure, that the nominal latency between receiving the data from the server, and the arrival of the data at the receiver is below 1 second (plus what is being optionally set in the latency value). If the latency time is longer, the receiver may experience prolonged or even failed startups.

If the latency is not deterministic (meaning, that the client cannot announce this to the server through the latency parameter), then one may choose to only use the 'eph' command.

4.2 Response format

Upon successful reception of a client's request, the server will return data and terminate the connection.

The data is split in two parts:

- A header containing ASCII data, describing the data that follows
- A data section with 8-bit raw binary data or ASCII data, depending on the request.

In case of a successful, fully authorized request, a typical response from the server looks as follows:

```
u-blox a-gps demo server (c) 1997-2008 u-blox AG\n <1>
Content-Length: 2392\n <2>
Content-Type: application/ubx\n <3>
\n <4>
<<5> 2392 bytes of binary day data>
```

Everything within "<" and ">" is not being sent by the server, but is added here for descriptive purposes. The "\n" indicates the newline character.

1. This is the welcome message from the server. Please note that this message may change.
2. The Content-Length line will tell how many bytes of data will be sent in the data section
3. The Content-Type line describes what format the data is in. In case of a successful request, this will be in "application/ubx" format
4. An empty line indicates the end of the header, and the start of the data section
5. In the data section, the indicated number of raw 8-bit data bytes will be sent. The client shall forward this data unmodified to the GPS receiver. Please note that in cases of "full" and "aid" commands, the received data shall be forwarded to the GPS receiver immediately, as the data contains approximate knowledge of time. Delays of 1 second or above may result in degraded startup performance of the receiver. In case an immediate forwarding is not possible, specify the expected latency as part of the request, using the "latency" parameter.

If an error happens, the server will respond slightly different:

```
u-blox a-gps demo server (c) 1997-2010 u-blox AG\n <1>
Content-Length: 38\n <2>
Content-Type: text/plain\n <3>
\n <4>
error: no approximate position given\n <5>
```

Everything within "<" and ">" is not being sent by the server, but is added here for descriptive purposes. The "\n" indicates the newline character.

1. This is the welcome message from the server
2. The Content-Length line will tell how many bytes of data will be sent in the data section
3. The Content-Type line describes what format the data is in. In case of a failure request, this will be in "text/plain" format
4. An empty line indicates the end of the header, and the start of the data section
5. In the data section, the indicated number of ASCII data bytes will be sent. The client shall **not** forward this to the receiver, as it is an error message

The following error messages are currently delivered by the server:

Error Message:	**Reason**
`no command given`	There was no command in the request
`invalid command`	A command was requested but is not supported
`authorization failed`	- Username or password is missing - The username is not valid - The password does not match to the username
`no approximate position given`	Neither lat and lon, nor ex, ey and ez values were given in the request

Please note that other error messages may be sent, or may change. However, error messages will always be indicated by a Content-Type text/plain and contain a description of the cause of error.

5 Best practices

5.1 Use DNS, don't use hardcoded IP numbers

u-blox reserves the right to change the IP address of the AssistNow Online service at any time, or even dynamically. Therefore, always use the fully qualified host name, using proper DNS lookups to resolve the IP address from the hostname as defined in Appendix B.

In case of an IP address change, the TTL values (time to live) of the DNS entries are steered in a way such that the clients always have a valid IP address despite a change-over.

5.2 Observe the Content-length information

GPS is a dynamic system. Since new satellites enter and exit the constellation, the return size that the server delivers is not fixed.

Therefore, the client needs either to have a dynamically sized buffer (sized according to the Content-length header field), or to crop the received data in case only fixed sized buffers are available.

5.3 Don't scan the whole planet

Some users wishing to retrieve the whole data set that AssistNow has available have been observed doing this by scanning a geographic region step-wise, in order to collect the current ephemeris data for every point in their region or even over the whole planet.

This is not a sensible approach, as it uses up bandwidth without any real benefit. u-blox reserves the right to refuse service to those performing such extensive use.

There are three options to make this smarter:

5.3.1 Use the *ephfull* command

The *ephfull* command will return all ephemeris for all satellites in a single reply. Since the ephemeris is only changing every two hours, this command should not be used more often than every two hours

5.3.2 Change to Assist Now Online Premium

u-blox offers a premium version of the AssistNow Online Service. The premium service operates in an encrypted, authenticated and 24/7 monitored manner, and allows fail-safe operation. u-blox supports premium service customers to set up their own proxy servers, which then deliver aiding data to end-user devices according to a proprietary customer protocol. In addition to the whole set of raw GPS data, the premium service also delivers satellite location prediction data. This allows a customer's proxy servers to determine satellite visibility at any location on the planet and hence only deliver the ephemeris which is visible there.

5.3.3 Maximum parameters

In case neither of the above two options are possible; the following maximum parameters must be obeyed.

- Minimum Grid Size: 20 degrees Longitude x 25 degrees Latitude (i.e. a maximum of 126 transactions) i.e. step through Longitude -160, -140, …. -20, 0, +20, … +140, +160, +180 degrees, and Latitude -75, -50, -25, 0, +25, +50, +75 degrees.

  - There is no real benefit in retrieving a smaller grid, it would only increase the bandwidth consumed

- Repeat this scan at most every 15 minutes.
  - Since the GPS ephemeris information does not change too frequently, 30 minutes is a sufficiently high update rate
- Don't perform this scan from more than two different servers at the same time.
- Allow at least 0.5 second between individual request whilst scanning

u-blox reserves the right to discontinue the service if any of the above rules are not respected.

A Sample client implementation

The following code implements a typical client. It connects to the server, sends its request, receives the data and (if of application type UBX) sends that data to a GPS receiver via a serial line.

The sample code is written in Perl language and has been tested on a Linux system. Please see your Operating System and Programming Language manuals for information on generic Serial Port- and Socket-Programming. u-blox also has a C-code sample available upon request.

```
#!/usr/bin/perl -w

# Package for Serial Port handling:
use Device::SerialPort;
# Package for Socket connections:
use IO::Socket;

################################
# INITIALIZATION               #
################################


################################
# RS232 settings               #
################################

$port = '/dev/ttyS0';
$baud = 57600;

################################
# SERVER settings              #
################################
$aiding_server = 'agps.u-blox.com';
$aiding_port   = 46434;
$user = 'foo@bar.com';
$pwd = 'whatever';
$lat = 47.28;
$lon = 8.56;
$acc = 1000;

$request = "cmd=aid;user=$user;pwd=$pwd;lat=$lat;lon=$lon;pacc=$acc\n";

# open the serial port, using the Device::SerialPort methods
my $gps = Device::SerialPort->new ($port) || die "Can't open $port: $!";

# apply settings to the serial port
$gps->baudrate($baud)  || die "fail setting baudrate";
$gps->parity("none")   || die "fail setting parity";
$gps->databits(8)      || die "fail setting databits";
$gps->stopbits(1)      || die "fail setting stopbits";
$gps->handshake("none")|| die "fail setting handshake";
$gps->write_settings   || die "no settings";

################################
# OPEN CONNECTION TO SERVER    #
################################

print "Connecting to server $aiding_server:$aiding_port\n";

$agps = IO::Socket::INET->new("$aiding_server:$aiding_port")
    || die "unable to contact $aiding_server: $!\n";


################################
# SUBMIT REQUEST               #
################################
```

```
print $agps $request;



################################
# RECEIVE RESPONSE HEADER      #
################################

while ($_ = $agps->getline)
{
    # $_ now contains one ASCII line from the server

    if (/^\s*\n$/)  # if we get an empty line
    {
      last; # jump out of loop
    } else {
      # Look whether it has t
      if ( $_ =~ /Content-(\S+):\s*(\S+)/)
      {
          $header{$1} = $2;
      } else {
          # server sends something we don't need.
      }
    }
}


#################################
# LOOK AT HEADER, DECIDE ACTION #
#################################

# If Content Type is text/plain, we dump data to
# the terminal[must be error or something else
# the GPS receiver does not understand]

if (defined $header{'Type'} && ($header{'Type'} eq "text/plain"))
{
    while($_ = $agps->getline)
    {
      print $_;
    }

}

# If the Content-Type is is application/ubx
# we read in that full chunk of data
# and send it to the receiver right away
if (defined $header{'Type'} && ($header{'Type'} eq "application/ubx"))
{
    # read a number of bytest
    if ( read($agps,$ubxdata,$header{'Length'}) != $header{'Length'})
    {
      die "wrong number of bytes received\n";
    }

    # say what we're doing
    printf "Aiding Data: %s bytes \n",$header{'Length'} ;

    # send it to the u-blox receiver
    $gps->write($ubxdata);

    # ... and what until the Serial TX buffer is emptied
    $gps->write_drain();
}

# close the GPS receiver port
undef $gps;
# and the socket
undef $agps;
```

B Server settings

The following connection settings shall be used

Server Name	agps.u-blox.com
Server Port	46434
Protocol	TCP or UDP
Username / Password	Please contact agps-account@u-blox.com

Note that the server

- Imposes timeouts on the connections. Idle connections will be closed by the server
- Imposes rate-limitations (only a certain number of connections per client are allowed). It will automatically slow down clients if the rates are exceeded
- Will log all transactions, to facilitate debugging.

Revision history

Revision	Date	Name	Status / Comments
-	17/07/2009	tgri	Initial release
A	02/12/2009	damm	Revised version
B	07/12/2009	tgri	Revised version for new u-blox CI
C	08/04/2010	sbut	Revised version for new AssistNow naming convention

Contact

For complete contact information visit us at www.u-blox.com

u-blox Offices

North, Central and South America

u-blox America, Inc.

Phone: +1 (703) 483 3180
E-mail: info_us@u-blox.com

Regional Office West Coast:

Phone: +1 (703) 483 3184
E-mail: info_us@u-blox.com

Technical Support:

Phone: +1 (703) 483 3185
E-mail: support_us@u-blox.com

Headquarters
Europe, Middle East, Africa

u-blox AG

Phone: +41 44 722 74 44
E-mail: info@u-blox.com
Support: support @u-blox.com

Asia, Australia, Pacific

u-blox Singapore Pte. Ltd.

Phone: +65 6734 3811
E-mail: info_ap@u-blox.com
Support: support_ap@u-blox.com

Regional Office China:

Phone: +86 10 68 133 545
E-mail: info_cn@u-blox.com
Support: support_cn@u-blox.com

Regional Office Japan:

Phone: +81 03 5775 3850
E-mail: info_jp@u-blox.com
Support: support_jp@u-blox.com

Regional Office Korea:

Phone: +82 2 542 0861
E-mail: info_kr@u-blox.com
Support: support_kr@u-blox.com

Regional Office Taiwan:

Phone: +886 2 2657 1090
E-mail: info_tw@u-blox.com
Support: support_tw@u-blox.com