Application Note: OpenGTS Device Communication Server (DCS)

Scope
AT110, AT200, AT210, AT240

Overview
OpenGTS is an open source project that provides web-based GPS tracking of devices from various manufacturers, including the following Astra Telematics devices:

- AT110
- AT200
- AT210
- AT240

Details on OpenGTS can be found here:

http://www.opengts.org/

Related Documents
The following documents are recommended reading to accompany this document:

- AT110, AT200, AT210 & AT240 User Guides
- AT110, AT200, AT210 & AT240 Installation Guides

These documents can be obtained from:

http://www.gps-telematics.co.uk/downloads.htm

New OpenGTS Implementations
OpenGTS must be installed and configured in order to receive data from an Astra Telematics device. Instructions for installation and configuration are detailed in the OpenGTS Installation and Configuration Manual which can be obtained from the OpenGTS website.

Before compiling the OpenGTS source code and initialising the data tables, configure the Astra DCS. Once the Astra DCS is configured, start the server. Devices should be configured and added to an account. This document will guide you on how to carry out these steps.

Existing OpenGTS Implementations
Update to the latest OpenGTS installation to obtain the latest Astra Device Communication Server. Configure the server as described in the next section and then compile OpenGTS before starting the Astra DCS.

Configuring the Astra DCS

The Astra DCS has several configuration parameters. These can be found in the file *dcservers.xml* in the OpenGTS directory. We recommend that these parameters are left as default values, except for the listening ports, which you may prefer to change. The ports on which the Astra DCS listens for incoming data packets are specified in the "ListenPorts" tag within <DCServer name="astra"> tag:

```
<!-- === Listen ports -->
<ListenPorts
    tcpPort="31090"
    udpPort="31090"
    />
```

The default port is 31090. If required, the listening ports can be changed to fit the requirements of your runtime environment. The tracking devices will need to be configured to transmit data to the same port as the ones the server uses to listen for incoming data packets. To do this, refer to the Device Configuration section later in this document.

The listening ports must be open through the firewall for the remote device to send data to the Astra server. To achieve this, check the NAT settings of your router and make sure no other application is listening on that port. After reading this guide, once you are running the server you can check if the ports are open. This can be done with websites such as http://www.yougetsignal.com/tools/open-ports/ , for example.

If packet acknowledgement is required, it must be sent from the same IP address to which the remote device sent the data packet. If your server responds to more than one IP address, then the Astra server listener must be bound to the same IP address/interface used by the remote tracking devices. This is set in the top-level *"dcservers.xml* " file, on the "DCServerConfig" tag, "bindAddress" attribute.

```
<!-- ========================
<DCServerConfig
    bindAddress=""
    backlog=""
    includeDir="dcservers"
    >
```

Note that the IP address of the server must be static. If necessary, contact your ISP or check No-IP services or similar, which provide remote access with dynamic DNS.

Device Configuration

Refer to the relevant product User Guide for complete descriptions of the configuration commands.

a) Configuring the socket

Set the server's IP or hostname using the following command

$IPAD,<ip_address>

Set the server's TCP/UDP listening port with the following command

$PORT,<port_number>

Where: <ip_address> and <port_number> are substituted for your server IP address (or hostname) and port number. Note that the brackets <> should not be included.

b) Configuring the protocol

The Astra DCS supports the following protocols:

- Protocol C (legacy)
- Protocol K
- Protocol M
- Protocol V (CAN and OBD data)

The protocol should be selected dependent upon the device as follows:

Device	Protocol	Command
AT200	K	$PROT,6
AT210	K	$PROT,6
AT110 non-CANBus	M	$PROT,8
AT240 non-CANBus	M	$PROT,8
AT110 CANBus	V	$PROT,14
AT240 CANBus	V	$PROT,14

c) Setting the communication mode

The Astra DCS supports the following communication modes:

- TCP (login disabled)
- UDP

The MODE must be set to TCP or UDP as required.

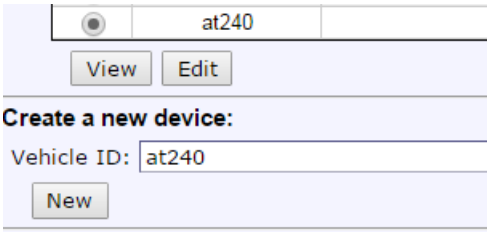
Mode	Command
TCP (login disabled)	$MODE,4
UDP	$MODE,5

The Astra DCS will send acknowledgement to the tracking device on successful decoding of received packets. The device time limit from sending a packet to receipt of the acknowledgement is set by the command:

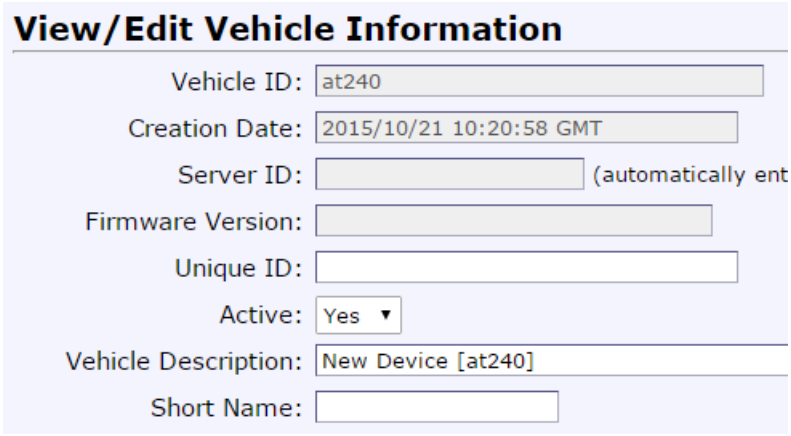
$TCPT,<seconds>

Adding a Device to OpenGTS

To add a device in OpenGTS, log into the server and click on the Administration /Vehicle Admin menu. Once there, insert an identifier in the Vehicle ID field. Vehicle ID can be a name.



Once introduced, click on new, select the recently created Vehicle and click on Edit. The following window will appear.



Fill in Unique ID with the device's 15 digit IMEI number, which is printed on the device label (or query using the $IMEI command) and the Vehicle Description field. The rest of the fields can be completed as desired.

Device Data

Data sent by the device is stored in the OpenGTS EventData database table. The following fields are populated by each report

Field	Units
deviceID	15 digit IMEI
timestamp	time and date in seconds from 00:00:00 1 January 1970
statusCode	See table below
latitude	decimal degrees (+/- 90)
longitude	decimal degrees (+/- 180)
speedKPH	km/h
heading	Degrees
altitude	Metres
distanceKM	km (journey distance travelled)
odometerKM	km (lifetime odometer)
rawData	See descriptions below

The possible OpenGTS values for the statusCode are listed below:

Open GTS status code
STATUS_IGNITION_ON
STATUS_IGNITION_OFF
STATUS_MOTION_EXCESS_IDLE
STATUS_MOTION_IN_MOTION
STATUS_MOTION_DORMANT
STATUS_MOTION_IDLE
STATUS_MOTION_START
STATUS_MOTION_MOVING
STATUS_MOTION_HEADING
STATUS_MOTION_EXCESS_SPEED
STATUS_PANIC_ON
STATUS_POWER_ON
STATUS_POWER_OFF
STATUS_INPUT_STATE
STATUS_LOW_BATTERY
STATUS_EXCESS_ACCEL
STATUS_EXCESS_BRAKING
STATUS_EXCESS_CORNERING
STATUS_IMPACT
STATUS_TOWING_START
STATUS_GEOFENCE_ARRIVE
STATUS_GEOFENCE_DEPART
STATUS_INITIALIZED
STATUS_HEARTBEAT
STATUS_QUERY
STATUS_BREACH_ON
STATUS_NOTIFY

CANBus Data

To support CANBus data, the following line must be uncommented in the OpenGTS config.conf file

```
startupInit.EventData.CANBUSFieldInfo=true
```

If the OpenGTS database already exists, update it using the command

```
dbAdmin.pl -tables=ca  (Linux)
dbConfig.bat -tables:ca(Windows)
```

When the device is configured for protocol V the CANBus data sent by the device is stored in the OpenGTS EventData database table. The following fields are populated as indicated for basic, start and stop reports:

Field	Units	Report type
engineRpm	RPM	Basic/start/stop
throttlePos	%	Basic/start/stop
engineLoad	%	Basic/start/stop
workDistanceKM	Km	Basic/start/stop
coolantTemp	degrees Centigrade	Basic/start/stop
fuelLevel	%	Basic/start/stop
fuelTotal	litres	Basic/start/stop
engineOnHours	hours	Stop
fuelTrip	litres per 100km	Stop
malfunctionLamp	N/A	Basic/start/stop (OBD only)

Additional Data

The flag addRawData can be set true in the TrackClientPacketHandler.java class to enable the Astra server to insert additional data in to the rawData field. Please refer to the relevant protocol description documentation. Setting addRawData to false will leave the rawData field empty.

For protocol K an example of the contents would be

R=000040;S=0101;P=12.2V;B=100%;D=01;A1=0.0V;M=0km/h;X=0,0;Y=0,0;Z=0,0;I=0s;Q=A4;G=00

where

R = Reason Code (24 bits)
S = Status Code (16 bits)
P = External power (Volts)
B = Battery level (Percentage)
D = Digital input/output status and state changes (8 bits)
A1 = External ADC1 reading (0-5 Volts)
M = Maximum journey speed (km/h)
X = Accelerometer X-axis max, min readings (m/s/s * 10)
Y = Accelerometer Y-axis max, min readings (m/s/s * 10)

Z = Accelerometer Z-axis max, min readings (m/s/s * 10)
I = Journey Idle Time (Seconds)
Q = Signal quality (MS nibble: GSM [0-15], LS nibble: number of GPS satellites in use)
G = GeoFence Event

For protocol M an example of the contents would be

R=000080;S=0100;P=12.4V;B=100%;D=000100;A1=0.0V;A2=0.0V;M=0km/h;X=3,0;Y=0,2;Z=0,8;I=14s;Q=C8;G=00

where

R = Reason Code (24 bits)
S = Status Code (16 bits)
P = External power (Volts)
B = Battery level (Percentage)
D = Digital input/output status and state changes (24 bits)
A1 = External ADC1 reading (0-5 Volts)
A2 = External ADC2 reading (0-15 Volts)
M = Maximum journey speed (km/h)
X = Accelerometer X-axis max, min readings (m/s/s * 10)
Y = Accelerometer Y-axis max, min readings (m/s/s * 10)
Z = Accelerometer Z-axis max, min readings (m/s/s * 10)
I = Journey Idle Time (Seconds)
Q = Signal quality (MS nibble: GSM [0-15], LS nibble: number of GPS satellites in use)
G = GeoFence Event

For protocol V an example of the contents would be

R=000004;S=0408;P=12.6V;B=95%;D=000000;A1=0.0V;A2=0.0V;M=0km/h;X=0,1;Y=0,2;Z=2,5;I=0s;Q=B9;G=00;CAN:W=80km/h;FS=0000;OS=0000;CS=0000;TD=0m;avW=0km/h;avE=0RPM;avA=0%;avL=0%;AW=0kg;CT=0s;SD=0km;RE=0s;DM=0km

where

R = Reason Code (24 bits)
S = Status Code (16 bits)
P = External power (Volts)
B = Battery level (Percentage)
D = Digital input/output status and state changes (24 bits)
A1 = External ADC1 reading (0-5 Volts)
A2 = External ADC2 reading (0-15 Volts)
M = Maximum journey speed (km/h)
X = Accelerometer X-axis max, min readings (m/s/s * 10)
Y = Accelerometer Y-axis max, min readings (m/s/s * 10)
Z = Accelerometer Z-axis max, min readings (m/s/s * 10)
I = Journey Idle Time (Seconds)
Q = Signal quality (MS nibble: GSM [0-15], LS nibble: number of GPS satellites in use)

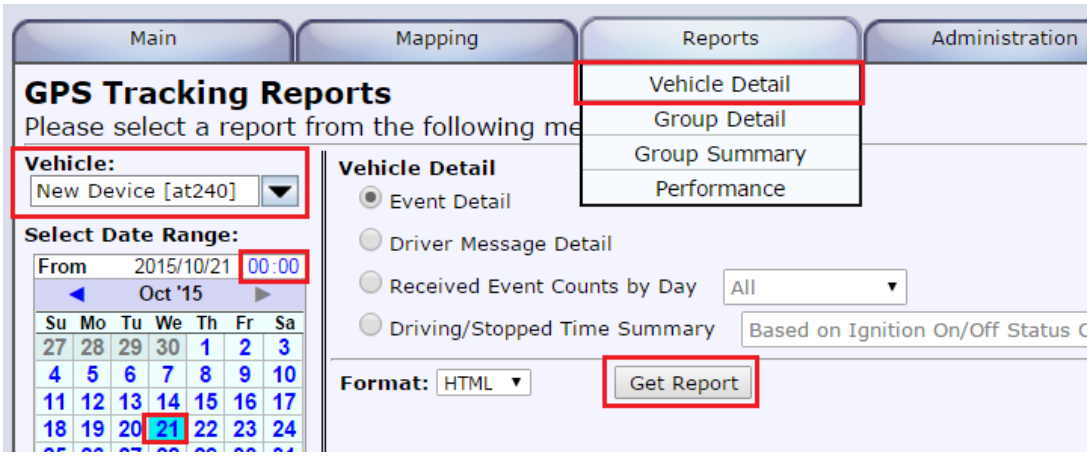
G = GeoFence Event

CANBus data following CAN:

W = Wheel Based Speed (km/h)
FS = FMS status (16 bits)
OS = OBD status (16 bits)
CS = CANBus Monitoring Status (16 bits)
TD= Total Vehicle Distance (m)
avW = Wheel Based Speed Average (km/h)
avE = Engine Speed Average (RPM)
avA = Accelerator Position Average (%)
avL = Engine Load Average (%)
AW = Axle Weight (kg)
CT = Journey Cruise Control Time (s)
SD = Service Distance (km)
RE = Run Time Since Engine Start (s)
DM = Distance Travelled With MIL On (km)

Viewing reports

To view a device's report, click on the Reports tab and select Vehicle Detail. Select the vehicle on the left hand side of the UI, a starting time or date and click on get report.



The mapping tab can also be used to view the device's location on a map.

Running the server

Once everything is configured, type in CMD the following command to run Astra's DCS:

%GTS_HOME%\bin\runserver astra

Further information

See the README.txt file in src/org/opengts/servers/astra (UNIX/Linux) or src\org\opengts\servers\astra (Windows).