astra telematics

Application Note: CANBus Commissioning

Scope
AT220B, AT240, AT110

Overview
Our devices can be connected to the CANBus network in a vehicle to obtain information from the electronic control units that communicate using that network. This application note describes device configuration for use on the CANBus network.

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

- AT220B, AT240 & AT110 User Guides
- AT220B, AT240 & AT110 Installation Guides
- AT220B Protocol L – CANbus
- Astra Protocol V – CAN and OBD

The first two documents can be obtained from:

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

The Protocol Description documents are available on request by emailing support@gps-telematics.co.uk

Compatibility
The CAN-H and CAN-L connections on the device must be connected to the vehicle CANBus network appropriately. Please refer to the User Guide for the appropriate device for pin numbers and wire colours.

Our devices support J1939 Fleet Management System (FMS) Standard. Extended 29 bit CAN identifiers are expected and the bit rate on the bus should be 250 kbit/s.

For an OBD interface the device supports bit rates of 250/500 kbit/s and the CAN identifiers may be standard 11 bit or extended 29 bit identifiers.

Protocol L only supports FMS data. When configured for Protocol V our devices can automatically detect whether the CANBus is FMS or OBD and configure the CANBus interface automatically. Otherwise, the settings can be configured manually using the CANC command:

$CANC,< silent mode>,<bit rate index>,<extended CAN ID>,<interface type>

The silent mode option operates as described in the following table:

silent mode	Description
0	silent mode off (uses dominant ACK bits. Device will acknowledge received messages)
1	silent mode on (uses recessive ACK bits. Device will not acknowledge received messages)

The bit rate index is in the range 0-2 and represents an actual bit rate as given in the following table:

Bit rate index	Bit rate
0	125 kbit/s
1	250 kbit/s
2	500 kbit/s

The extended CAN ID option selects 11 bit or 29 bit CAN identifiers for CAN transmissions as follows:

Extended CAN ID	Description
0	Standard 11 bit identifiers
1	Extended 29 bit identifiers

The interface type is set as follows:

Interface type	Description
0	Auto-detect
1	FMS
2	OBD (Note that the device will transmit on the CANBus)

If the auto-detect process is selected in protocol V and faults are indicated on the vehicle then change to the appropriate fixed FMS/OBD mode.

FMS CANBus monitoring

CANBus data is reported as part of protocol L,N or V and the following parameters should be presented on the network by the vehicle control units:

Parameter Group Number (PGN)	Description
0xFEF1 (65265)	Cruise Control/Vehicle Speed
0xF003 (61443)	Electronic Engine Controller #2
0xFEE9 (65257)	Fuel Consumption
0xFEFC (65276)	Dash Display
0xF004 (61444)	Electronic Engine Controller #1
0xFEEA (65258)	Vehicle Weight
0xFEE5 (65253)	Engine Hours
0xFEEC (65260)	Vehicle Identification
0xFDD1 (64977)	FMS-standard Interface
0xFEC1 (65217)	High Resolution Vehicle Distance
0xFEC0 (65216)	Service Informations
0xFE6C (65132)	Tachograph
0xFEEE (65262)	Engine Temperature 1
0xFEF5 (65269)	Ambient Conditions

Parameter Group Number (PGN)	Description
0xFE6B (65131)	Driver's Identification
0xFEF2 (65266)	Fuel Economy
0xFDA4 (64932)	PTO Drive Engagement
0xFD09 (64777)	High Resolution Fuel Consumption

Note that for a Vehicle Identification greater than 8 bytes long the following parameters will be used to present the information:

Parameter Group Number (PGN)	Description
0xECFF (60671)	Broadcast Announce Message
0xEBFF (60415)	Transport Protocol – Data Transfer

OBD CANBus monitoring

OBD data is reported as part of protocol V and the following parameters may be presented on the network by the vehicle electronic control units:

Parameter ID	Description
0x01	Malfunction indicator lamp status and number of diagnostic trouble codes to display
0x04	Engine load
0x05	Engine coolant temperature
0x0C	Engine RPM
0x0D	Vehicle speed
0x11	Throttle position
0x1F	Run time since engine start
0x21	Distance travelled with malfunction indicator lamp on
0x2F	Fuel level input

Configuring Events

FMS CANBus events can be configured to generate reports using the following command:

$CANM,<canbus_event_mask>

where the mask bits are set to 1 to enable event triggers and cleared to disable event triggers. The mask bits are described in the following table:

Status	Bit	Default
Brake switch – pedal released	0	0
Brake switch – pedal depressed	1	0
Cruise control – switched on	2	1
Cruise control – switched off	3	1
PTO – Off / Disabled	4	1
PTO – Set	5	1
PTO – Not Available	6	1
Vehicle Direction – Forward	7	1
Vehicle Direction – Reverse	8	1
Vehicle Speed – Overspeed	9	1

Status	Bit	Default
Vehicle Speed – No Overspeed	10	1
Reserved	11	0
Reserved	12	0
Reserved	13	0
Reserved	14	0
Reserved	15	0

Therefore, the default CANM setting is 2044 (0x7FC).

FMS/OBD Configuring of Event Reporting Thresholds

The CANBus event reporting thresholds can be configured by setting the relevant parameters using the commands described below:

Engine load

$ELRT,<engine_load_high_threshold>
$ELHT,<engine_load_high_hold_timeout>
$ELIT,<engine_load_high_inhibit_timeout>

Parameter	Description
engine_load_high_threshold	value which when reached or exceeded by the reported value will generate a report. A setting of 0 turns off event reporting for this threshold.
engine_load_high_hold_timeout	the time (in seconds) for which the reported engine load must exceed the engine_load_high_threshold setting in order for an event to be reported
engine_load_high_inhibit_timeout	the time (in seconds) following an engine load high event for which another event cannot be reported

The engine load is reported on a scale of 0-125 percent of the operational range of FMS and 0-100 for OBD.

RPM

$RPRT,<rpm_high_threshold>
$RPHT,<rpm_high_hold timeout>
$RPIT,<rpm_high_inhibit timeout>

Parameter	Description
rpm_high_threshold	value which when reached or exceeded by the reported value will generate a report. A setting of 0 turns off event reporting for this threshold.
rpm_high_hold_timeout	the time (in seconds) for which the reported RPM must exceed the rpm_high_threshold setting in order for an event to be reported
rpm_high_inhibit_timeout	the time (in seconds) following RPM high event for which another event cannot be reported

The RPM is reported divided by 32 on a scale of 0-250 (to represent 0-8000 rpm) for FMS and 0-255 (0-8160) for OBD.

Throttle position

$THRT,<rpm_high_threshold>
$THHT,<rpm_high_hold timeout>
$THIT,<rpm_high_inhibit timeout>

Parameter	Description
throttle_high_threshold	value which when reached or exceeded by the reported value will generate a report. A setting of 0 turns off event reporting for this threshold.
throttle_high_hold_timeout	the time (in seconds) for which the reported throttle position must exceed the throttle_high_threshold setting in order for an event to be reported
throttle_high_inhibit_timeout	the time (in seconds) following a throttle high event for which another event cannot be reported

The throttle position is reported on a scale of 0-250 to represent position 0-100% for protocol L. In protocol V throttle position is always reported on a scale of 0-100%.

To summarise, the ranges for the event threshold parameters are given in the table below:

Parameter	Minimum	default value	Maximum
engine_load_high_threshold	1	90	125
engine_load _hold_timeout (secs)	1	30	65535
engine_load _inhibit_timeout (secs)	1	60	65535
rpm_high_threshold	1	4000	8000
rpm_hold_timeout (secs)	1	30	65535
rpm_inhibit_timeout (secs)	1	60	65535
throttle_high_threshold	1	75	100
throttle_hold_timeout (secs)	1	30	65535
throttle_inhibit_timeout (secs)	1	60	65535

Diagnostics

FMS

In the first instance it is useful to check if any J1939 messages are being received. Issuing the command

$DIAG,101

will produce a list of parameter group numbers each with their associated *approximated* repeat rate (in milliseconds). For example:

```
F003 60
FEE9 1070
F004 100
FE6C 60
FEEE 1070
FEF5 1070
$DIAG,OK
```

From this we can see that F003 and FE6C are being received every 50 ms, F004 every 100mS and FEE9,FEEE and FEF5 every 1000mS

To view data as required for protocol L send the command

$DIAG,102

which will result in a response similar to the following:

```
CAN Bus: speed=0 rpm=86 accel=73 eng_load=255 dist=0 temp=43 stat=0x0 mst=0x124
         fuel_lev=0 wt=0 eng_hrs=0 fuel_used=42085 (0) cr_ctrl_t=0 serv_dist=0
         VIN=ABC123
```

To view the last received data for an individual parameter enter the following command:

$DIAG,103

which will result in a response similar to the following:

```
F003 FF 4F FF FF FF FF FF FF
FEE9 FF FF FF FF D5 07 00 00
F004 FF FF FF 40 1B FF FF FF
FE6C FF FF FF FF FF FF C0 09
FEEE 35 50 FF FF FF FF FF FF
FEF5 FE FF FF 19 00 23 FF FF
```

which shows the parameter and the 8 bytes of data last received for that parameter.

FMS and OBD

To continually view the identifiers and optionally data on the CANBus send the command

$DIAG,104,d,t

where d is 0,1 or 2 and controls the data output as follows:

D	Output
0	Suppressed
1	Identifiers only
2	Identifiers and data

The t option can be 0 or non-zero and controls the display of a relative millisecond counter as follows:

T	Output
0	No timestamp is displayed
>0	Relative millisecond timestamp is displayed

For example with $DIAG,104,1 output will look similar to the following

```
CF00300
CFE6C00
CF00400
18FEEE00
```

```
CF00300
18FEF500
CFE6C00
18FEE900
18FEF600
18FECA00
CF00300
CFE6C00
```

For example with $DIAG,104,2 output will be in the format <CAN ID> <CAN data> look similar to the following

```
CF00300 FF 35 FF FF FF FF FF FF
CFE6C00 FF FF FF FF FF FF C0 10
CF00400 FF FF FF 80 22 FF FF FF
18FEEE00 43 50 FF FF FF FF FF FF
CF00300 FF 35 FF FF FF FF FF FF
18FEF500 FE FF FF 19 00 23 FF FF
CFE6C00 FF FF FF FF FF FF C0 10
18FEE900 FF FF FF FF 45 1F 00 00
18FEF600 FF 70 FF FF FF FF FF FF
18FECA00 03 FF 00 00 00 00 00 00
CF00300 FF 35 FF FF FF FF FF FF
CFE6C00 FF FF FF FF FF FF 80 10
```

NOTE: If DIAG 104 outputs 29 bit CANBus identifiers when connected to FMS it is bits 23-8 that represent the parameter group numbers used in the FMS standard.

With $DIAG,104,1,1 output will be in the format <millisecond timestamp>: <CAN ID> and look similar to the following

```
8640: CF00300
8650: CFE6C00
8690: CF00300
8700: CFE6C00
8730: CF00400
8750: CF00300
8760: CFE6C00
8800: CF00300
8810: CFE6C00
8840: CF00400
8850: CF00300
8860: CFE6C00
8910: CF00300
8920: CFE6C00
```

With $DIAG,104,2,1 output will be in the format <millisecond timestamp>: <CAN ID> <CAN data> and look similar to the following

```
210: CF00300 FF 63 FF FF FF FF FF FF
220: CFE6C00 FF FF FF FF FF FF 00 00
250: CF00400 FF FF FF 40 71 FF FF FF
260: CF00300 FF 63 FF FF FF FF FF FF
270: CFE6C00 FF FF FF FF FF FF 00 00
320: CF00300 FF 63 FF FF FF FF FF FF
330: CFE6C00 FF FF FF FF FF FF 00 00
360: CF00400 FF FF FF 40 71 FF FF FF
370: CF00300 FF 63 FF FF FF FF FF FF
380: CFE6C00 FF FF FF FF FF FF 00 00
```

The timestamp starts from 0 when it is first turned on can be reset to 0 by issuing the the command $DIAG,104,1,1 or $DIAG,104,2,1

Protocol V auto-detect

The CANBus standard (FMS or OBD) can be automatically detected in protocol V. The auto-detect process can be manually started using the command $DIAG,106. For fixed CANBus configuration in protocol V see the description of the $CANC command in the Compatibility section above.

If an FMS interface is detected the debug output will contain the following

```
* FMS detected *
```

If OBD is detected the debug output will contain output similar to the following

```
canbus: ----------------------
canbus: 29 bit IDs, 500 kbit/s
canbus: ----------------------
canbus: ECU: 18DAF110
canbus: ----------------------
canbus: Supported PIDs:
canbus: 1
canbus: 2
canbus: 3
.
.
.
```

OBD PIDs supported

When connected to CANBus via the OBD interface the PIDs supported can be requested as bitmasks using the command $DIAG,107

OBD fuel tank capacity

When connected to CANBus via the OBD interface the fuel tank capacity is set using the command $DIAG,110,<tank_capacity> where <tank_capacity> is in litres.