



dataTaker CANgate CAN Frequently Asked Questions

Table of Contents

Introduction	2	Do I need a terminating network on the CAN ports?	11
General overview	3	Are CAN and OBD connectors available as accessories?	11
What is CANgate?	3	Which versions of CANBUS protocols are supported?	11
Can I log data directly within the CANgate?	3	Are parameters such as vehicle speed or temperature available?	11
Data loggers	3	Is support included for FMS-Standards?	11
Does CANgate work with all dataTaker data loggers?	3	Can I Filter the CAN data available for logging?	12
Which dataTaker data logger models work with CANgate?	3	What is the difference between broadcast and polled CAN parameters?	12
What do I need to know to interface models other than the DT80 range products?	4	Can I 'poll' CAN data from another device?	12
How do I use CANgate with a DT50/500/600 range logger?	4	Can I 'write' data or CAN values to a CAN device?	12
How do I use CANgate with a DT800 logger?	7	How fast can I log CAN data?	12
Mechanical & hardware	8	How many CAN data points can be logged or returned?	12
What mounting options are available?	8	CAN sensors	13
What is the vibration rating for a CANgate?	8	Is it possible to connect directly to CAN sensors?	13
Drop tested to what?	8	GPS data	13
What is the MTBF/MTTF for the CANgate?	8	Can a GPS be connected to the CANgate?	13
Are CAN and OBD connectors available as accessories?	8	Is a particular type or brand of GPS required?	13
Battery/power	9	Can I power a GPS unit from the CANgate?	13
Why doesn't the CANgate include a plug pack/power supply?	9	Firmware	13
Is the unit self-powered?	9	Is the firmware upgradeable?	13
Will settings be lost when power is removed?	9	Programming & commands	14
Can I power from an external battery?	9	Do I have to learn the programming text language?	14
Can I power from a data logger?	9	What debugging tools are available?	14
How do I tell if external power is available?	9	How do I determine if I have CAN bus comms errors?	14
Can I power a GPS unit from the CANgate?	9	How can I learn more about programming?	15
LED indicators	9	CAN protocols	15
What do the LEDs indicate?	9	Which CANBUS protocols are supported?	15
Host communications	10	Where can I get information on OBD-II parameters?	15
Programming tools available?	10	Where can I get information on J1939 parameters?	15
Can I connect to the CANgate remotely?	10	Where can I get information on NMEA 0183?	15
Can I connect over the web?	10	Where can I get information on NMEA 2000?	15
Is the RS232 port Isolated?	10	Does CANgate work with systems which follow the CAN 2.0 B spec?	15
Do I need to upgrade DeLogger 4 for use with CANgate?	10	Reverse engineering	16
Do I need to upgrade DeTransfer?	10	How do I go about reverse engineering CANBUS messages?	16
Can I view real-time data?	10	How do I work out the CANgate program if I am using the CANgate directly without a logger?	18
DeLogger seems to 'hang' when sending CANgate programs to a DT80, what can I do?	10	How do I measure fault codes from an ECU?	21
CAN communications	10	How do you add and remove J1939 parameters to the DeLogger configuration file?	21
How many CAN ports are provided?	10		
How do I connect to multiple ports on the CANgate as there is only one connector?	10		

Introduction

Please note the CANgate CAN to ASCII gateway is now obsolete.

General overview

What is CANgate?

The dataTaker CANgate CAN to ASCII gateway filters and converts CAN (Controller Area Network) and GPS (Global Positioning System) data to serial ASCII data. It has been designed to capture the real-time data available on CAN and GPS networks (e.g. temperatures, RPM, brake/throttle settings, latitude, longitude, etc.) and pass that information on to either a dataTaker data logger or host computer system.

Two CAN interface ports are included, which allow connection of two independent CAN networks. CANgate's versatile configuration options allow you to select the parameters of interest, apply statistical functions (average, minimum, maximum), and control the format in which data are returned. CANgate supports widely used protocols such as ISO-15765 and SAE-J1939, as well as raw CAN frames.

Many CAN networks are used in vehicle and marine applications where positional information can be an important part of the data collected. CANgate includes a GPS interface to allow integration of the positional and other information available from GPS systems.

The dataTaker DeLogger host software application supports the use of CANgate in conjunction with the dataTaker DT80 range of data loggers. CAN and GPS parameters of interest may be selected from a predefined database of SAE J1939, OBD-II, and NMEA-0183 (GPS) parameters. This database may be modified or extended by the user, allowing custom parameters and protocols to be supported. DeLogger allows easy integration of CAN and GPS data collected via CANgate with traditional analog and digital measurements.

When interfaced to a host computer system directly, CANgate can be configured by simple ASCII commands. This allows a terminal software application such as dataTaker DeTransfer or HyperTerminal to be used to configure and collect data from CANgate.

Example application areas include trucks, cars, trains, mining vehicles, and stationary diesel engine equipment.

Can I log data directly within the CANgate?

No, the CANgate is not a data logger, it is capable of filtering MIN/MAX/AVG data from broadcast CAN messages, but only temporarily stores this information.

Data loggers

Does CANgate work with all dataTaker data loggers?

Yes, all ASCII interface dataTaker units are capable of use with CANgate. The preferred and simplest combination to use with CANgate is the DT80 range and DeLogger 4 host software. Other combinations required detailed knowledge and use of the ASCII text programming language for both the data logger and CANgate.

Which dataTaker data logger models work with CANgate?

The CANgate was developed and is supported primarily for the DT80 range. Other models capable of use with CANgate include, DT5x, DT5xx, DT6xx, & DT800 products. Note that host software support for CANgate is only directly provided by DeLogger for the DT80/CANgate combination.

What do I need to know to interface models other than the DT80 range products?

Use of the CANgate with models other than the DT80 range (or with the DT80 but without DeLogger) requires knowledge and use of ASCII text commands for both the data logger and the CANgate. For the DT800, it is possible to use the Serial Sensor port or RS232 Host port for connection to CANgate; for other models, the Host port must be used. If using the Host port, the CANgate must be programmed to directly output ASCII commands that can be used by the dataTaker; Example: 1CV=100.

How do I use CANgate with a DT50/500/600 range logger?

These loggers do not have a programmable serial sensor port, so the data returned from the CANgate must be formatted as a 'command' to load CAN data values into Channel Variables. Programming of the CANgate must be via a text interface although you can use DeLogger to generate an initial program that must then be modified to suit use with a DT50/500/600 range logger.

The steps are as follows:

1. Setup a program for the CANgate using the DT80 program builder in DeLogger When you 'right click' on the program and select 'Show Program'. The program will look something like this.

```
'JOB=JOB1
'COMPILED=2007/04/10 15:22:20
'TYPE=dt80
DT=\d
001: BEGIN"JOB1"
002: CATTN
003: 'Spans and polynomial declarations
004: 'Thermistor declarations
005: 'Switches declarations
006: 'Parameter declarations
007: 'Global declarations
008: RS1S
009: ' Initialise port that CANgate is connected to
010: PS=RS232,57600,N,8,1,HWFC
011: ' Initialise the CANgate device
012: IF(1SERIAL("{\e{VERBOSE OFF;VERSION^M}%f",2)<1.23)"CANgate not found or firmware version less than 1.23"
013: ' Initialise CANgate ports
014: 1SERIAL("{CONNECT 1 250;SETADDR 1 0^M}",W) 'J1939
015: 1SERIAL("{CONNECT 2 0;SETADDR 2 0^M}",W)
016: 1SERIAL("{GPSBAUD 19200^M}",W) 'Garmin GPS18 5Hz
017: ' Initialise selected CAN parameters
018: 1SERIAL("{BEGIN^M}",W)
019: 1SERIAL("{1 GPS \034GPGGA\034 2 D^M}",W)
020: 1SERIAL("{2 GPS \034GPGGA\034 2 M^M}",W)
021: 1SERIAL("{3 GPS \034GPGGA\034 4 D^M}",W)
022: 1SERIAL("{4 GPS \034GPGGA\034 4 M^M}",W)
023: 1SERIAL("{5 GPS \034GPGGA\034 6 N^M}",W)
024: 1SERIAL("{6 RECVJ 1 61444 4 5 256 3 0 FORMAT un 0.125 0 \034%%.3f\n\034 ^M}",W)
025: 1SERIAL("{END^M}",W)
026: 'schedule definition
027: RA("B:",ALARMS:OV:100KB,DATA:OV:1MB)1S LOGONA GA
028: 1SERIAL("{\e{RP 1^M}%f",1,"LatD~Deg")
029: 1SERIAL("{\e{RP 2^M}%f",1,"LatM~Min")
030: 1SERIAL("{\e{RP 3^M}%f",1,"LonD~Deg")
031: 1SERIAL("{\e{RP 4^M}%f",1,"LonM~Min")
032: 1SERIAL("{\e{RP 5^M}%d",1,"Fix Quality~State")
033: 1SERIAL("{\e{RP 6^M}%f",2,"EEC1Engine Speed~rpm")
034: END
'end of program file
```

2. You need to copy the 1SERIAL lines before the RA command

```
014: 1SERIAL("{CONNECT 1 250;SETADDR 1 0^M}",W) 'J1939
015: 1SERIAL("{CONNECT 2 0;SETADDR 2 0^M}",W)
016: 1SERIAL("{GPSBAUD 19200^M}",W) 'Garmin GPS18 5Hz
018: 1SERIAL("{BEGIN^M}",W)
019: 1SERIAL("{1 GPS \034GPGGA\034 2 D^M}",W)
020: 1SERIAL("{2 GPS \034GPGGA\034 2 M^M}",W)
021: 1SERIAL("{3 GPS \034GPGGA\034 4 D^M}",W)
022: 1SERIAL("{4 GPS \034GPGGA\034 4 M^M}",W)
023: 1SERIAL("{5 GPS \034GPGGA\034 6 N^M}",W)
024: 1SERIAL("{6 RECVJ 1 61444 4 5 256 3 0 FORMAT un 0.125 0 \034%.3f\\n\034 ^M}",W)
025: 1SERIAL("{END^M}",W)
```

3. Remove the 1SERIAL part of the line i.e. (Including the parts after the ^M)

```
CONNECT 1 250;SETADDR 1 0
CONNECT 2 0;SETADDR 2 0
GPSBAUD 19200
BEGIN
1 GPS \034GPGGA\034 2 D
2 GPS \034GPGGA\034 2 M
3 GPS \034GPGGA\034 4 D
4 GPS \034GPGGA\034 4 M
5 GPS \034GPGGA\034 6 N
6 RECVJ 1 61444 4 5 256 3 0 FORMAT un 0.125 0 \034%.3f\\n\034
END
```

4. Replace the \034 with the Quote character " and %% with %

```
CONNECT 1 250;SETADDR 1 0
CONNECT 2 0;SETADDR 2 0
GPSBAUD 19200
BEGIN
1 GPS "GPGGA" 2 D
2 GPS "GPGGA" 2 M
3 GPS "GPGGA" 4 D
4 GPS "GPGGA" 4 M
5 GPS "GPGGA" 6 N
6 RECVJ 1 61444 4 5 256 3 0 FORMAT un 0.125 0 "%.3f\\n"
END
```

5. We need to add a measurement rate (in this case every 2000ms) to the CANgate configuration and add a channel variable 'preamble' to the data. Note for GPS information we use %s for formatting GPS strings (as far as the CANgate is concerned GPS characters are strings so scaling etc will not work on the GPS information!) The scaling and appropriate '%' characters are given for the CANbus information.

```
CONNECT 1 250;SETADDR 1 0
CONNECT 2 0;SETADDR 2 0
GPSBAUD 19200
BEGIN
1 GPS "GPGGA" 2 D 2000 FORMAT "1CV=%s\\n"
2 GPS "GPGGA" 2 M 2000 FORMAT "2CV=%s\\n"
3 GPS "GPGGA" 4 D 2000 FORMAT "3CV=%s\\n"
4 GPS "GPGGA" 4 M 2000 FORMAT "4CV=%s\\n"
5 GPS "GPGGA" 6 N 2000 FORMAT "5CV=%s\\n"
6 RECVJ 1 61444 4 5 256 3 0 2000 FORMAT un 0.125 0 "6CV=%.3f\\n"
END
```

6. We now need to connect directly to the CANgate using DeTransfer and the above characters are sent to the CANGATE (or if the above lines are sent via Delogger and the SSdirect method via a DT80 then the \\ will need to be replaced by \ ...)

7. The CANgate should now return the readings as follow

```
1CV=-37
2CV=-54.48425
3CV=145
4CV=14.83269
5CV=0
6CV=818.125
```

8. The CANgate can now be connect to a 500 series logger (using the standard PC cable) running the following code:

```
BEGIN
RA2S
1CV("LatD~Deg")
2CV("LatM~Min")
3CV("LonD~Deg")
4CV("LonM~Min")
5CV("Fix Quality")
6CV("EEC1Engine Speed")
END
```

The 500 series logger will now be recording CAN & GPS information.

How do I use CANgate with a DT800 logger?

Because DeLogger only supports CANgate for the DT80, you must use an alternative approach to configuration and programming. There are several approaches you can take.

Program the units individually: Refer to the CANgate manual for text based programming information allowing formatted data to be returned to the DT800 serial sensor port. Program the DT800 using DeTransfer or DeLogger to accept the returned data and log it appropriately.

Start with the DT80 CANgate builder, and modify the resulting program to suit the DT800.

The steps are as follows:

1. Setup a program for the CANgate using the DT80 program builder in DeLogger. When you 'right click' on the program and select 'Show Program' The program will look something like this.

```
'JOB=JOB1
'COMPILED=2007/04/10 15:22:20
'TYPE=dt80
DT=\d
001: BEGIN"JOB1"
002: CATTN
003: 'Spans and polynomial declarations
004: 'Thermistor declarations
005: 'Switches declarations
006: 'Parameter declarations
007: 'Global declarations
008: RS1S
009: ' Initialise port that CANgate is connected to
010: PS=RS232,57600,N,8,1,HWFC
011: ' Initialise the CANgate device
012: IF(1SERIAL("\e{VERBOSE OFF;VERSION^M}%f",2)<1.23)"CANgate not found or firmware version less than 1.23"
013: ' Initialise CANgate ports
014: 1SERIAL("{CONNECT 1 250;SETADDR 1 0^M}",W) 'J1939
015: 1SERIAL("{CONNECT 2 0;SETADDR 2 0^M}",W)
016: 1SERIAL("{GPSBAUD 19200^M}",W) 'Garmin GPS18 5Hz
017: ' Initialise selected CAN parameters
018: 1SERIAL("{BEGIN^M}",W)
019: 1SERIAL("{1 GPS \034GPGGA\034 2 D^M}",W)
020: 1SERIAL("{2 GPS \034GPGGA\034 2 M^M}",W)
021: 1SERIAL("{3 GPS \034GPGGA\034 4 D^M}",W)
022: 1SERIAL("{4 GPS \034GPGGA\034 4 M^M}",W)
023: 1SERIAL("{5 GPS \034GPGGA\034 6 N^M}",W)
024: 1SERIAL("{6 RECJV 1 61444 4 5 256 3 0 FORMAT un 0.125 0 \034%.3f\n\034 ^M}",W)
025: 1SERIAL("{END^M}",W)
026: 'schedule definition
027: RA("B:",ALARMS:OV:100KB,DATA:OV:1MB)1S LOGONA GA
028: 1SERIAL("\e{RP 1^M}%f",1,"LatD~Deg")
029: 1SERIAL("\e{RP 2^M}%f",1,"LatM~Min")
030: 1SERIAL("\e{RP 3^M}%f",1,"LonD~Deg")
031: 1SERIAL("\e{RP 4^M}%f",1,"LonM~Min")
032: 1SERIAL("\e{RP 5^M}%d",1,"Fix Quality~State")
033: 1SERIAL("\e{RP 6^M}%f",2,"EEC1Engine Speed~rpm")
034: END
'end of program file
```

2. The following lines would need to be changed

```
012: IF(1SERIAL("\e{VERBOSE OFF;VERSION^M}%f",2)<1.23)"CANgate not found or firmware version less than 1.23"
```

to

```
1SERIAL("\e{VERBOSE OFF;VERSION^M}%f[1CV]",2)  
IF(1SERIAL(1CV<1.23)"CANgate not found or firmware version less than 1.23"
```

and the measurement channels changed to:

```
027: RA("B:",ALARMS:OV:100KB,DATA:OV:1MB)1S LOGONA GA  
028: 1SERIAL("\e{RP 1^M}%f",1,"LatD~Deg")  
029: 1SERIAL("\e{RP 2^M}%f",1,"LatM~Min")  
030: 1SERIAL("\e{RP 3^M}%f",1,"LonD~Deg")  
031: 1SERIAL("\e{RP 4^M}%f",1,"LonM~Min")  
032: 1SERIAL("\e{RP 5^M}%d",1,"Fix Quality~State")  
033: 1SERIAL("\e{RP 6^M}%f",2,"EEC1Engine Speed~rpm")
```

changed to

```
RA1S LOGONA GA  
1SERIAL("\e{RP 1^M}%f[1CV]",1) 1CV("LatD~Deg")  
1SERIAL("\e{RP 2^M}%f[2CV]",1) 2CV("LatM~Min")  
1SERIAL("\e{RP 3^M}%f[3CV]",1) 3CV("LonD~Deg")  
1SERIAL("\e{RP 4^M}%f[4CV]",1) 4CV("LonM~Min")  
1SERIAL("\e{RP 5^M}%d[5CV]",1) 5CV("Fix Quality~State")  
1SERIAL("\e{RP 6^M}%f[6CV]",2) 6CV("EEC1Engine Speed~rpm")
```

Mechanical & hardware

What mounting options are available?

No mounting holes or brackets are currently supplied. User must determine most appropriate method for individual application. Examples would be the use of Cable Ties (Zip Ties), Velcro, Double sided adhesive tape, and fabricated brackets.

What is the vibration rating for a CANgate?

The CANgate unit is not currently rated.

Drop tested to what?

This is drop tested only in packaging for shipping purposes.

What is the MTBF/MTTF for the CANgate?

We do not provide a figure for this. Our products are designed to be robust and reliable, and we have no hesitation in offering a 3-year warranty to back up that claim.

Are CAN and OBD connectors available as accessories?

No, there is a DE9 to screw terminal adaptor supplied with the CANgate to connect wires from the various ports or you can make up your own customized cable.

Battery/power

Why doesn't the CANgate include a plug pack/power supply?

A plug pack is not normally needed for most applications. Power for the CANgate will usually come from the supply for the CAN network devices, typically 12 or 24 V. If there is no power available from the CAN network, then you can power the CANgate from the supply for the logger or a dedicated supply such as a plug pack.

Is the unit self-powered?

No, the CANgate is externally powered.

Will settings be lost when power is removed?

Flash memory holds configuration so that if power is lost or removed, the configuration is not lost. This can easily be used in applications where the system only runs (and is only powered) when an engine or CAN data source is running/ powered.

Can I power from an external battery?

Yes, although most CAN systems have a 12 or 24 V power system available. It is expected that a battery will not be required in most cases. External power supply range is +10 to +30 VDC supply.

Can I power from a data logger?

Yes, but this is model dependent as to voltages and methods available.

How do I tell if external power is available?

Visually-coloured indicator LEDs show Power status and CAN1/ CAN2/ GPS activity.

Can I power a GPS unit from the CANgate?

Yes, the CANgate provides a +5 Vdc (200 mA Max) output which can be used to power some GPS units such as Garmin GPS 18 5Hz.

LED indicators

What do the LEDs indicate?

The 6 coloured indicator LEDs show Power status and CAN1/ CAN2/ GPS / RS232 activity..

- Power LED(red) on while CANgate is powered
- Host Tx LED(red) flashes when characters are transmitted to host port
- Host Rx LED(green) flashes when characters are received from host port
- CAN1 Rx LED(blue) flashes when a CAN frame is successfully received from CAN port 1
- CAN2 Rx LED(red) flashes when a CAN frame is successfully received from CAN port 2
- GPS Rx LED(green) flashes when characters are received from GPS port

Note: The CAN LEDs will only flash on receipt of CAN frames which match CANgate's current hardware filter settings. If CANgate has not yet been programmed to receive CAN data then the LEDs will not flash, even if there is traffic on the connected bus.

Host communications

Programming tools available?

Programming from DeLogger is supported via the DT80 Program Builder with selection of CANgate measurements from drop down lists. These lists are based on XML files that can be customised by the user to create application specific channels if required.

NOTE: DeLogger provides combined CANgate/DT80 (range) data logger programming, not standalone programming of the CANgate unit.

Text programming of the CANgate unit is possible via the Host RS-232 port for advanced users. In this case, a terminal program such as DeTransfer or HyperTerminal can be used.

Can I connect to the CANgate remotely?

Yes, via serial Modem or Ethernet when connected locally to a DT80 data logger.

Can I connect over the web?

Yes, you can connect to via a web connected DT80 data logger.

Is the RS232 port Isolated?

No.

Do I need to upgrade DeLogger 4 for use with CANgate?

Yes, the minimum version of DeLogger designed for use with CANgate is DeLogger4 Version 4 Release 1.

Do I need to upgrade DeTransfer?

No, but note DeTransfer does not include any specific support for either the CANgate or DT80.

Can I view real-time data?

Yes, but only at the speed that the connected data logger can handle data.

DeLogger seems to 'hang' when sending CANgate programs to a DT80, what can I do?

For reliable transfer of a DeLogger CANgate program to a DT80, ensure the CANgate unit is connected to the DT80 and is powered during the transfer.

CAN communications

How many CAN ports are provided?

The CANgate has two CAN ports allowing connection to two independent CAN buses running different protocols and BUS speeds.

How do I connect to multiple ports on the CANgate as there is only one connector?

There is a single DE9 female connector that has dedicated pins for connection to two separate CAN buses, an RS232 connection to a GPS unit, as well as power input and output. You can either use the supplied DE9 to screw terminal adaptor to connect wires from the various ports to the CANgate or you can make up your own customised splitter cable that connects the relevant signals.

The CAN1/CAN2/GPS/Power connector has the following pinout:

Pin	Signal
1	5V out
2	Ground
3	CAN 1 High
4	GPS Rx
5	CAN 1 Low
6	CAN 2 High
7	CAN 2 Low
8	GPS Tx
9	10 to 30V in

Do I need a terminating network on the CAN ports?

Not usually. Terminating networks are resistors fitted to the end of the last CAN bus connection. Terminators help to prevent CAN signal errors, without termination errors that can cause unpredictable results. The CANgate includes a **weak** 2.6 KOhm termination network. See the CANgate user's manual for a detailed description on the need for and use of termination networks.

Are CAN and OBD connectors available as accessories?

No, there is a DE9 to screw terminal adaptor supplied with the CANgate to connect wires from the various ports, or you can make up your own customized cable.

Which versions of CANBUS protocols are supported?

The CANgate has support for RAW CAN messages plus J1939, OBD-II, and ISO-14230 protocols. Physical layer support is for high speed CAN as defined in ISO 11898-2.

Are parameters such as vehicle speed or temperature available?

Yes, if the CAN system connected also supports published protocols and standards such as ISO 14230-3/OBD-II or SAE J1939-71, this information should be readily available. Alternatively, if the information is contained in unpublished CAN messages, the user will require detailed information on the message format to allow appropriate CANgate programming.

Is support included for FMS-Standards?

Not at time of release, however as FMS-Standards are closely related to J1939, it should be possible to create a custom XML database for DeLogger. Power/Advanced users should be able to setup the CANgate appropriately.

About FMS-Standards: Some major truck manufacturers have agreed to give third parties access to vehicle data via the CAN bus. This includes DaimlerChrysler, MAN, Scania, DAF Trucks, IVECO, Volvo Trucks, and Renault Trucks. They have designed a common interface as an open standard which will be supported by these manufacturers.

NOTE: Each manufacturer may make small changes to the implementation of the FSM-Standard for their particular needs.

For further information on The FMS-Standards, please visit the official FMS-Standards website.

Can I Filter the CAN data available for logging?

Yes, the CANgate can filter desired parameters and data from broadcast messages on the CAN bus. Example: Extract a single or particular data items from a large number of broadcast items. 150 memory slots are available for filtering broadcast messages.

NOTE: For supported protocols such as OBD-II and J1939 only. For other filtering the user will require knowledge of meaning of messages.

What is the difference between broadcast and polled CAN parameters?

Broadcast parameters are transmitted regularly on a CAN network by the ECU that senses and generates the information. This information can be read by the CANgate without disturbing the operating of the network. This is commonly known as “sniffing” data on the network. Broadcasting information can be an efficient way of transmitting parameter data, particularly when there are multiple consumers of the data. The SAE J1939 defines many parameters that are broadcast on the network.

Polled parameters must be requested. The CANgate can poll parameters when the particular protocol is known, such as SAE J1939 or OBD-II. Polled parameters require the CANgate to send and receive CAN frames to obtain the data. As such it is an intrusive process on the CAN network as it adds more traffic to the network. Care should be taken to avoid overloading a CAN network with too many requests for polled parameters.

Can I ‘poll’ CAN data from another device?

Yes, the CANgate can poll for parameters by sending out request messages on the CAN bus for the supported protocols OBD-II and J1939. Example: Request Engine Speed or Temperature using OBD-II protocol.

Can I ‘write’ data or CAN values to a CAN device?

Yes, but note that all messages sent are in RAW format, and it is up to the programmer to encode the correct data in the message.

How fast can I log CAN data?

The effective logging speed of CAN data is set by the data logger capability, not the CANgate itself. The CANgate is capable of data reduction on higher speed CAN messages allowing good data resolution for most general purpose tasks.

How many CAN data points can be logged or returned?

For broadcast messages, a maximum of 150 data slots is currently available when used with the logger in DeLogger.

CAN sensors

Is it possible to connect directly to CAN sensors?

Yes, if you have details of the CAN messages and protocol supported by the sensor it may be connected directly to the CANgate unit.

GPS data

Can a GPS be connected to the CANgate?

Yes, the CANgate allows connection to GPS and other NEMA-0183 devices for collection of positional and other NEMA-0183 data.

Is a particular type or brand of GPS required?

No, any RS232 serial connected GPS receiver supporting NEMA-0183 should work with the CANgate. Our recommendation is to use a product from a major brand such as the Garmin products on our Third Party FAQ page. The standard communications rate of NMEA-0183 devices is 4800 baud. However, a number of GPS systems available will operate at considerably higher communications rates (Garmin GPS 18 5Hz, for example). The GPS communications rate on the CANgate can be set by using the GPSBAUD command i.e. GPSBAUD 19200 will set the comms rate of the GPS port on the CANgate to 19,200.

Can I power a GPS unit from the CANgate?

Yes, the CANgate provides a +5 Vdc (200 mA Max) output which can be used to power some GPS units such as Garmin GPS 18 5Hz.

Firmware

Is the firmware upgradeable?

Yes. Refer to the CANgate user's manual for details on how to upgrade the firmware.

Programming & commands

Do I have to learn the programming text language?

No, most straightforward CAN data logging tasks can be programmed using the DeLogger 4 program builder functions.

What debugging tools are available?

The DT80 range products support a serial sensor direct mode (SSDIRECT) that allows you to send commands and receive results directly to and from the CANgate via the dataTaker data logger. This means you can communicate directly with the CANgate without having to unplug it from the logger. DeLogger can enter this mode from within its text window. Once communicating directly with the CANgate, you can use the built-in commands such as SNOOP and SNOOPJ to examine existing traffic on a CAN or GPS network.

How do I determine if I have CAN bus comms errors?

When communicating with a CAN bus, you may find requested data such as OBD-II generates errors. This is usually due to poor shielding or incorrect termination on the CAN bus. To confirm that you have CAN bus communications problems, you can use the CANgate STATS command.

Use the Text window in DeLogger to enter the STATS command. A DT80 needs to be 'connected' to DeLogger, and the Text window also needs to be 'connected' to the DT80.

- Halt the DT80 by entering H
- Click the appropriate SSDIRECT [1->] or [2->] button to connect directly to the CANgate.
- Type STATS the CANgate should now display the results of the STATS command. An example is shown below.
- When you have completed the task, you need to exit SSDIRECT mode and enter G to start the DT80 again.

Example of data returned by the stats command:

```
HOST: Tx:1158 Rx:9856 bytes Dropped Tx:0 Rx:0 Errors:21
GPS:   Tx:0 Rx:40600 bytes Dropped Tx:0 Rx:0 Errors:3
CAN1: Tx:0 Rx:0 frames Dropped Tx:0 Rx:0
      Errors Warning:0 Bus:0 ArbLost:0
CAN2: Tx:184 Rx:184 frames Dropped Tx:0 Rx:0
      Errors Warning:0 Bus:0 ArbLost:6
Sys:   RQST dropped:0 Proc ovfl:0 Except: 0/0
```

Explanation of the STATS command:

```
STATS { CLEAR }
```

This diagnostic command displays communications statistics for each port (host, GPS, CAN1, CAN2). These include:

- Volume of traffic sent (Tx) and received (Rx), measured in bytes for the serial ports, and in frames for the CAN ports
- Number of transmit bytes/frames that could not be sent (Dropped Tx). This may occur if the slot sample rates are set too fast, so that the volume of traffic required to be sent exceeds the available bandwidth on the port, or if the quality of the communications on the BUS is poor. For example, for an OBDII command, for every received piece of data there is a transmitted piece of data so for poor communications there will be more Tx packets than Rx packets!
- Number of bytes/frames that could not be received and/or processed (Dropped Rx). This may occur if the slot sample rates are set such that CANgate cannot keep up with the rate of incoming messages.

- Number of bytes/frames which were discarded due to errors. These may be due to electrical noise or incorrect bit rate settings. Note that for CAN ports, “arbitration lost” (ArbLost) errors are a normal part of operation and simply indicate that a CANgate transmission was pre-empted by a higher-priority message.
- Number of CANgate system errors, including RQST/RQSTJ queue overflows and processing overflows (occasions when the configured slot sample rates could not be met due to excessive processing load)
- An indication as to whether an exception (serious CANgate error that required a firmware restart) has occurred.
- If the CLEAR parameter is specified, then all statistics values are reset.

How can I learn more about programming?

The CANgate user’s manual contains detailed information on programming and using the CANgate.

DataTaker also conducts training workshops suitable for beginners or experienced users.

CAN protocols

Which CANBUS protocols are supported?

The device, as well as being able to read RAW CAN messages, supports the following protocols:

- CAN 2.0B PROTOCOL / ISO 11898-2 Data Link / Physical Layer
- ISO 14230-3 / OBD-II Diagnostic Services Application Layer
- ISO 15765-2 Network Layer
- SAE J1939-71 Vehicle Application Layer
- SAE J1939-21 Data Link / Network Layer
- NMEA-0183
- NMEA-2000 (This is currently untested .. however it is supposed to be the same as J1939 so it should work)

Where can I get information on OBD-II parameters?

SAE J1979 - E/E Diagnostic Test Modes

Wikipedia OBD-II PIDs article

Where can I get information on J1939 parameters?

SAE J1939

Where can I get information on NMEA 0183?

NMEA0183 reversed engineered public source 1

NMEA0183 reversed engineered public source 2

Where can I get information on NMEA 2000?

NMEA web site

Does CANgate work with systems which follow the CAN 2.0 B spec?

Yes, CANgate supports CAN 2.0B (which defines the basic standard and extended packet format). There are many different protocols which operate using CAN 2.0B (eg. J1939, ISO 14230 etc.). As a minimum, CANgate will support capture of raw CAN frames from a CAN2.0B device. Depending on which other protocols in use, additional functions may be available.

CAN can operate over a few different physical layer standards. However, CANgate only supports the ISO 11898-2/SAEJ2284 physical layer, sometimes referred to as “high speed CAN”, which is probably the most widely used. The important characteristics of this standard are:

- Two wire, 5V differential signalling
- Twisted pair cable, 120R characteristic impedance
- Bit rate: 10 kbps–1 Mbps

Reverse engineering

How do I go about reverse engineering CANBUS messages?

The CANgate can be quite effective as a tool to reverse engineer what CANBUS messages are available on a particular CANBUS. The following is a generalized procedure

- Connect the CANgate to the CANBUS you wish to investigate.
- Run DeLogger.
- Make a software connection between DeLogger and the logger the CANgate is connected to.
- Switch to the ‘Text’ window.
- Check the Text window is connected to the same logger. (The drop down box in the bottom left of the Text Window)
- `type:PS=57600,N,8,1,HWFC`
- Press the serial sensor direct mode button
`-type:connect 1 250`
- `type:Snoop 1`
- If no data is returned, then there are three possibilities. Firstly, CANBUS may not be operating. Alternatively, the CANgate may not be connected to the CANBUS. Another possibility is that the CANgate is set to the wrong communications rate.
- If you type: `Verbose ON`, then you will get a message telling you if the connection rate is wrong.
- Repeat with different data rates–10, 20, 50, 125, 250, 500, 1,000.
- When you have the correct connection rate type, set verbose mode to off `type:Verbose OFF`
- The snoop command returns all of the current messages available on the CANBUS
- Each message needs to be checked. If you can disconnect the device you are interested in when you do the snoop command, it should be the device that is now ‘missing’.
- If you are unable to disconnect the particular device, you will need to manipulate the device in some way to get it to the device to ‘change’.
- Each message can be examined in the following way:
`type:RECV 1 0x0200 1 8 all`
- Where 1 is the CANgate port we are using. 0x0200 is the address we are investigating. 1 to 8 are the bytes to be returned and we are going to return ALL messages.
- Either the message will stop when it is unplugged, or it will change when the particular sensor is manipulated.
- Warning: Some sensors require an initialization string to be sent to them before they will start transmitting. These are the exceptions rather than the rule. These sensors would need to be ‘insitu’ to successfully investigate. Sometimes the easiest method to investigate a particular sensor is to plug only that sensor into the CANgate. That way it is obvious when we use the snoop command what the address of that particular sensor is.
- Once you have established which address the device you need to establish which part of the message is being used, try to identify the bits involved in the manipulation.
- The RECV command can be used to return only specific bits. For Example
`type:RECV 1 0x200 1 2 all`
- This would return the first two bytes from the string. You then need to determine how the bytes are arranged and how they are scaled.
- The command would look something like this:
`type:RECV 1 0x0200 1 2 all format UM 1 0 “%f\n”`
- The variations to try would be UM UN SM SN

- You may also wish to return the number in different formats such as %f floating point, %d signed integer, %u unsigned integer and %s string
- By manipulating the sensor, you also need to set the scale and the offset.
- If you wish to display more messages at once, then you could use the following program:

```
BEGIN
  1 RECV 1 0x200 1 8 all format "0x200>%X\n"
  2 RECV 1 0x207 1 8 all format "0x207>%X\n"
  3 RECV 1 0x6F6 1 8 all format "0x6F6>%X\n"
END
```

Using a short 'program', you can investigate several messages at once. This program would return CANBUS messages 0x200, 0x207, & 0x6F6. Because of the limited communications rate of the CANgate, not ALL messages would be returned. If you want to check the number of messages missing, the STATS command will tell you the number of 'dropped' messages. The above program is useful sometimes to determine the start-up conversation between a sensor and the ECU. For example, usually the Talk and listen addresses are 8 apart (i.e. 7e0 and 7e8) in the case of most ECUs so the program would look something like:

```
BEGIN
  1 RECV 1 0x7e0 1 8 all format "0x7e0>%X\n"
  2 RECV 1 0x7e8 1 8 all format "0x7e8>%X\n"
END
```

How do I work out the CANgate program if I am using the CANgate directly without a logger?

The steps are as follows:

1. Setup a program for the CANgate using the DT80 program builder in DeLogger. When you 'right click' on the program and select 'Show Program', the program will look something like this. This program will record the GPS position, signal quality, and speed. Please note that a 'blank' field is returned as the speed whilst we have a poor GPS fix.

```
'JOB=JOB1
  'COMPILED=2007/06/11 05:49:42
  'TYPE=dt80
  DT=\d
001: BEGIN"JOB1"
002: CATTN
003: 'Spans and polynomial declarations
004: 'Thermistor declarations
005: 'Switches declarations
006: 'Parameter declarations
007: 'Global declarations
008: RS1S
009: ' Initialise port that CANgate is connected to
010: PS=RS232,57600,N,8,1,HWFC
011: ' Initialise the CANgate device
012: IF(1SERIAL("{\e{VERBOSE OFF;VERSION^M}%f",2)<1.23)"CANgate not found or firmware version less than 1.23"
013: ' Initialise CANgate ports
014: 1SERIAL("{CONNECT 1 0;SETADDR 1 0^M}",W)
015: 1SERIAL("{CONNECT 2 0;SETADDR 2 0^M}",W)
016: 1SERIAL("{GPSBAUD 19200^M}",W) 'Garmin GPS18 5Hz
017: ' Initialise selected CAN parameters
018: 1SERIAL("{BEGIN^M}",W)
019: 1SERIAL("{1 GPS \034GPGGA\034 6 N^M}",W)
020: 1SERIAL("{2 GPS \034GPGGA\034 2 D^M}",W)
021: 1SERIAL("{3 GPS \034GPGGA\034 2 M^M}",W)
022: 1SERIAL("{4 GPS \034GPGGA\034 4 D^M}",W)
023: 1SERIAL("{5 GPS \034GPGGA\034 4 M^M}",W)
024: 1SERIAL("{6 GPS \034GPVTG\034 1 N^M}",W)
025: 1SERIAL("{7 GPS \034GPVTG\034 7 N^M}",W)
026: 1SERIAL("{END^M}",W)
027: 'schedule definition
028: RA("B:",ALARMS:OV:100KB,DATA:OV:1MB)5S LOGONA GA
029: 1SERIAL("{\e{RP 1^M}%d",1,"Fix Quality~State",FF0)
030: 1SERIAL("{\e{RP 2^M}%f",1,"LatD~Deg",FF0)
031: 1SERIAL("{\e{RP 3^M}%f",1,"LatM~Min",FF6)
032: 1SERIAL("{\e{RP 4^M}%f",1,"LonD~Deg",FF0)
033: 1SERIAL("{\e{RP 5^M}%f",1,"LonM~Min",FF6)
034: 1SERIAL("{\e{RP 6^M}%f",1,"True Course~Deg",FF1)
035: 1SERIAL("{\e{RP 7^M}%f",1,"Ground speed~kph",FF1)
036: END
  'end of program file
```

2. The following lines are useful

```
014: 1SERIAL("{CONNECT 1 0;SETADDR 1 0^M}",W)
015: 1SERIAL("{CONNECT 2 0;SETADDR 2 0^M}",W)
016: 1SERIAL("{GPSBAUD 19200^M}",W) 'Garmin GPS18 5Hz
017: ' Initialise selected CAN parameters
018: 1SERIAL("{BEGIN^M}",W)
019: 1SERIAL("{1 GPS \034GPGGA\034 6 N^M}",W)
020: 1SERIAL("{2 GPS \034GPGGA\034 2 D^M}",W)
021: 1SERIAL("{3 GPS \034GPGGA\034 2 M^M}",W)
022: 1SERIAL("{4 GPS \034GPGGA\034 4 D^M}",W)
023: 1SERIAL("{5 GPS \034GPGGA\034 4 M^M}",W)
024: 1SERIAL("{6 GPS \034GPVTG\034 1 N^M}",W)
025: 1SERIAL("{7 GPS \034GPVTG\034 7 N^M}",W)
026: 1SERIAL("{END^M}",W)
```

But the 1SERIAL part needs to be removed because we are talking to the CANgate directly i.e.

```
CONNECT 1 0;SETADDR 1 0
CONNECT 2 0;SETADDR 2 0
GPSBAUD 19200
BEGIN
1 GPS \034GPGGA\034 6 N
2 GPS \034GPGGA\034 2 D
3 GPS \034GPGGA\034 2 M
4 GPS \034GPGGA\034 4 D
5 GPS \034GPGGA\034 4 M
6 GPS \034GPVTG\034 1 N
7 GPS \034GPVTG\034 7 N
END
```

The \034 can be replaced by the quote symbol " i.e.

```
CONNECT 1 0;SETADDR 1 0
CONNECT 2 0;SETADDR 2 0
GPSBAUD 19200
BEGIN
1 GPS "GPGGA" 6 N
2 GPS "GPGGA" 2 D
3 GPS "GPGGA" 2 M
4 GPS "GPGGA" 4 D
5 GPS "GPGGA" 4 M
6 GPS "GPVTG" 1 N
7 GPS "GPVTG" 7 N
END
```

We need to add a return rate in this case every 5,000 ms i.e.

```
CONNECT 1 0;SETADDR 1 0
CONNECT 2 0;SETADDR 2 0
GPSBAUD 19200
BEGIN
1 GPS "GPGGA" 6 N 5000
2 GPS "GPGGA" 2 D 5000
3 GPS "GPGGA" 2 M 5000
4 GPS "GPGGA" 4 D 5000
5 GPS "GPGGA" 4 M 5000
6 GPS "GPVTG" 1 N 5000
7 GPS "GPVTG" 7 N 5000
END
```

We also now format the return. Note the GPS string is treated as simple text by the CANgate and not as a floating point number as you would expect.

```
CONNECT 1 0;SETADDR 1 0
CONNECT 2 0;SETADDR 2 0
GPSBAUD 19200
BEGIN
1 GPS "GPGGA" 6 N 5000 FORMAT "Fix Quality=%s\n"
2 GPS "GPGGA" 2 D 5000 FORMAT "LatD~Deg=%s\n"
3 GPS "GPGGA" 2 M 5000 FORMAT "LatM~Min=%s\n"
4 GPS "GPGGA" 4 D 5000 FORMAT "LonD~Deg=%s\n"
5 GPS "GPGGA" 4 M 5000 FORMAT "LonM~Min=%s\n"
6 GPS "GPVTG" 1 N 5000 FORMAT "True Course~Deg=%s\n"
7 GPS "GPVTG" 7 N 5000 FORMAT "Ground speed~kph=%s\n"
END
```

If we are sending this program via DeTransfer, we will need to replace the \n with \\

How do I measure fault codes from an ECU?

To recover fault codes from an ECU requires a different programming approach because there will be a variable number of fault codes returned. The following is a code example that records the number of fault codes for each schedule, and when the number of fault codes change records, the fault codes present after the change. This code assumes that we have a CANgate connected to the logger via the serial sensor port. The CANBUS is connected to the CANgate via CAN port 2.

```
BEGIN
PS=57600,N,8,1,HWFC
FUNCTION1="Clr Flts"{XF}
1SERIAL("{CONNECT 2 500;SETADDR 2 0^M}",W) 'Orion HSCAN
RA1S
91CV(W)=(90CV>150CV)+(90CV<150CV)
' Read the number of current fault codes into 150CV
1SERIAL("\e{RQST 2 22_0200 FORMAT um 1 0 \\034%%.0d\\n\\034 ;RP^M}%d[150CV]",2,W)
150CV("NUM FAULT CODES")
' If the number of fault codes has changed then request the current list of fault codes from the ECU
ALARM(91CV>1.){[HA HB HC 1SERIAL("\e{rqst 2 18_0ff00 3 0;rp^M}",W) 701CV(W)=150CV}
RE1S:701CV LOGONE GE
701CV(W)=701CV-1
' Read each fault code into memory..
1SERIAL("%6s[1$]",=99CV,W)
1$("FLT CODES")
' OR Alternatively one could use the lines to read them as decimals..
' 1SERIAL("%6X[1CV]",=99CV,W)
' 1CV("FLT CODES")
IF(701CV<0.5){701CV(W)=0 GA GB GC 90CV(W)=150CV}
RFX 1SERIAL("\e{rqst 2 14_ff00;rp^M}",W)'Clears Fault codes
END
```

How do you add and remove J1939 parameters to the DeLogger configuration file?

DeLogger is supplied with pre-defined databases in XML (Extensible Markup Language) format, which can be readily edited to suit particular requirements or expanded to cover additional protocols or manufacturer-specific parameters. A detailed description of this process is included within the DeLogger on-line Help files under 'Advanced Topics'. Any standard text or XML editor (e.g. XMLnotepad, Oxygen, XML Spy, Stylus Studio) may be used to create/edit protocol XML files. Microsoft XMLnotepad is available from <http://www.microsoft.com/downloads/details.aspx?familyid=72d6aa49-787d-4118-ba5f-4f30fe913628&displaylang=en>. If this link does not work, search for XMLnotepad.

All protocol files are in the directory C:\Program Files\dataTaker\DeLogger\CANgate. Select and copy the standard J1939 file to use as a template. ('CANgate.Protocols.J1939.xml' type Ctrl-C Ctrl-V Right click on the file 'Copy of CANgate.Protocols.J1939.xml' and rename to say 'CANgate.Protocols.Custom.xml' or whatever is appropriate for your application.) Right Click on the file, and select 'Open With', and select XMLNotepad. If it is not available, then you will need to select 'Chose Program'. If XML Notepad is not available here, select 'Browse'. You should be able to find the program in the 'XML Notepad 2007' directory. You will need to select the XMLNotepad.exe file. When the file is opened in XML Notepad. You can select the 'records' and Cut, Delete, Copy, and Edit individual records.

In Australia:

For customer service, call 1300-735-292

To email an order, ordersau@thermofisher.com

In New Zealand:

For customer service, call 0800-933-966

To email an order, ordersnz@thermofisher.com

Find out more at thermofisher.com/datataker

ThermoFisher
S C I E N T I F I C