

## Application Note: CANBus Generic Can (\$CANG)

### Scope

CANBus capable devices

### Overview

\$CANG allows the user to configure the CANBus to listen for, report and react to specific CAN frames with specific content.

### Related Documents

The following documents are recommended reading to accompany this document:

- Astra Telematics Command Reference

### How to target J1939 PGNs

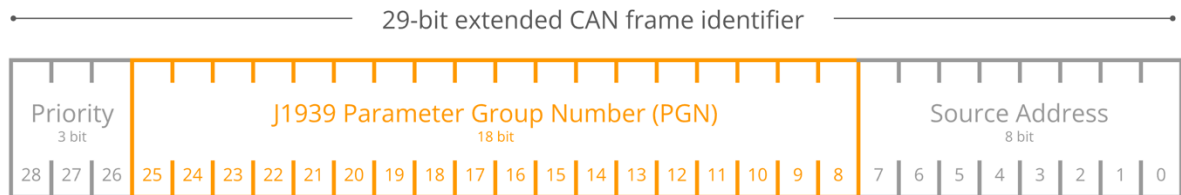
The \$CANG command structure allow the user to target CAN IDs at the rawest level by providing an ID and ID-mask parameter.

In short, the formula to target a J1939 PGN is as follows:

CAN ID: (PGN << 8)

Mask: 67108608d (0x3FFFF00h)

J1939 PGNs are 18-bits contained within a 29-bit extended CAN ID:



PGNs are in bits 8-25 (from 0). This means that to target CAN IDs on the bus containing that PGN, modifications to the CAN ID and CAN ID mask must be made.

Take this example of a J1939 PGN

PGN: 65262d (0xFEEEH)

To create the correct CAN ID and mask for this PGN, we must first shift the PGN to the left 8 bits (<< 8):

Original: 65262d (0xFEEEH)

Shifted: 16707072 (0xFEEEE00h)

Then, we must define a CAN ID mask that will allow this value to be compared to only the PGN bits in the CAN ID:

CAN ID Mask for J1939 PGNs: 67108608d (0x3FFFF00h)

So, the final \$CANG command to target this PGN (0xFEEEH) is:

`$CANG,<index>,16707072,67108608,1,0,0,0,0`

## How to differentiate messages with the same ID based on a MUX (multiplex) value within the payload

The \$CANG command structure allows further targeting of CAN messages beyond the ID, the user can also target messages based on data within the payload.

This is done using the *mux-start-bit*, *mux-bit-width* and *mux-value* parameters.

Say for example we want to target only when the third byte of the payload is equal to 171d (0xABh):

```
mux-start-bit: 16  
mux-bit-width: 8  
mux-value: 171
```

The final \$CANG command to target PGN 0xFEEEH when the third byte of the payload is equal to 171 is:

```
$CANG,<index>,16707072,67108608,1,16,8,171,0
```

## How to trigger reports when specified bits of the data payload change

The \$CANG command structure allows the user to define a 64-bit event mask that defines which bits of the data payload to check for changes to trigger a report.

This is done with the *event-mask* parameter.

Say for example, we want to trigger a report whenever there is a change in the first byte of the payload data:

```
event-mask: 18374686479671623680 (0xFF00000000000000)
```

The final \$CANG command to target PGN 0xFEEEH when the third byte of the payload is equal to 171 and to trigger a report whenever the value of the first byte changes is:

```
$CANG,<index>,16707072,67108608,1,16,8,171,18374686479671623680
```

Note: the device has an onboard report throttling system for \$CANG to prevent user-error in these configurations that could cause thousands of reports a second to be triggered.

There are two different throttles defined:

- Device will only allow a maximum of 5 reports to be triggered every 1000 milliseconds
- Device will only allow a maximum of 30 reports to be triggered every 60000 milliseconds