XGAUGE CODING INFORMATION

The purpose of this document is to explain the way data is used in the XGauge function in the ScanGaugeII.

TXD

The TXD field contains the command to transmit. It also seconds as the trip indicator when used to define a trip value as a gauge. If the first character is a space, this XGauge memory is not in use. Entries must be an even number of characters. If an odd number is input, a "0" is padded to it.

NON-CAN TXD

In non-CAN use, the first 3 bytes are used to designate the priority, send-to address and the ScanGaugeIIs address. These will be different depending on the required command. If the command is "legislated" or standard, the command will be different than if it is "non-legislated" or OEM defined. "non-legislated" are typically Mode 22 requests.

PWM Legislated	61	6A	F1
PWM non-Legislated	C4	10	F1
VPW, ISO & KWP Legislated	68	6A	F1
VPW, ISO & KWP non-Legislated	6C	10	F1

CAN TXD

The first 4 characters in CAN are the identifier. The following characters are the data to send. The ScanGaugeII looks for a response ID which has the high bit of the third character flipped from that of the transmit ID. The lower 3 bits of the third character are "don't care" bits for the receive. Further selection of the response in the RXF is used to screen the responses.

Example: A CAN mode $22 PID $4923 State-of-Charge for the Escape Hybrid would be:

0745224923

The 0745 is the ID to send for this request. The response must have an ID in the range of 0748 to 074F. In the case of the Escape Hybrid, the response ID would be $074D which would be accepted.

MODE

The MODE follows these bytes. The MODE determines what kind of action is needed. Most data requests are either MODE 01 ('legislated") or MODE 22 ("non-legislated").

NOTE: *The response will add 40 to the mode number of the request for its mode field.*

PIDs

Parameter Identifiers are used with MODE 01 and MODE 22 to determine what data is requested. MODE 01 PIDs are usually 2 characters (1 Byte) in length. MODE 22 PIDs are usually 4 characters (2 Bytes) in length.

CRC/Checksum

The ScanGaugeII adds any required CRC or Checksum to the request automatically. You don't have to do this. It is added when the command is sent and doesn't appear in TXD.

Special Values

There are cases where TXD contains something other than a command to send. For instance, it can be used to designate which trip function to show as a display. It can also contain "00" to indicate this is a valid gauge. Bits in the RXF indicate these special modes.

RXF

RXF is the receive filter used to determine if the received response is to our request. Also, RXF contains bits which indicate special ways to handle the data or use other things for the gauge data.

The first 3 bytes of the response for a non-CAN vehicle is similar to the command:

PWM Legislated	61	6B	xx
PWM non-Legislated	C4	11	xx
VPW, ISO & KWP Legislated	68	6B	xx
VPW, ISO & KWP non-Legislated	6C	11	xx

The "xx" would be the responding units address.

Note: The third character (byte 2) is increment by one from the TXD value. 6A in the TXD becomes 6B in the response.

There are 6 fields of 2 characters (1 byte) in RXF:

Offset 1	Match 1	Offset 2	Match 2	Offset 3	Match 3
00	00	00	00	00	00

The offset tells how many bytes into the response to look for a match of data. The Match tells what the byte value must be for a match.

You have to have a value for Offset 1 and Match 1 as a minimum for the RXF to look for a match. The other Offsets and Matches are optional.

For instance if we were looking for a Mode 22 PID 4923 for a VPW vehicle response, the response string would be: 6C6Bxx624923yy. "xx"" is the repsonding ECU address and "yyyy" is the data returned.

Each 2 characters is a byte in the response. If we use the mode and pid to filter the response, the RXF would be as follows: 046205490623

The breakdown is easier to see in the following table:

Offset 1	Match 1	Offset 2	Match 2	Offset 3	Match 3
04	62	05	49	06	23

Offset 1 says the 4th byte (starting with the first byte being 01) must be 62 (Match 1). Offset 2 says the 5th byte must be 49. Offset 3 says the 6th byte must be 23.

CAN Vehicles

CAN vehicles treat the response similar to the non-CAN vehicles. Internally, the ScanGaugeII treats the Identifier as a 3 byte value, even though in TXD it was only 2 bytes. By doing this, the RXF is the same for both CAN and non-CAN responses.

Special RXF values

Values in the first character (upper nibble) of the Offsets are used to signal special things to the ScanGaugeII. These are as follows:

First character=>	8	4	2	1
Offset 1	Is a TRIP gauge	Special1	Special2	Na
Offset 2	Value is 10x	Value is 100x	Value is ON/OFF	Value is HEX
Offset 3	Na	Na	Na	Na

If an 8 is the first character of Offset 1, the TXD value will be used to show a trip value as a gauge. This is explained further in the ScanGaugeII manual.

Offset 2 values are useful if the output value is to be a decimal value. By multiplying and dividing the response data so that the computed value is ten times the actual value and then using 8 for the first character of Offset 2, the value will have a single value after the decimal. By making it 100 times the numeric value and using 4 for the first character in Offset 2, there will be 2 digits past the decimal point.

If a single bit is pointed to by RXD and the first character of Offset 2 is made a 2, a value of "1" will display "ON" and a value of "0" will display "OFF".

If the first character of Offset 3 is 8, the value will be displayed as a hex-a-decimal value. It can be up to 4 characters (2 bytes).

RXD

RXD tells the ScanGaugeII where to locate the data value and how big it is. In this case we use bits.

Bits Offset from Start (00 = first bit)	Bits of length in the data (01 = 1 bit)
00	00

The entries are in hex. 00 is the first bit in the string, 08 is the first bit in the second byte, 10 is the first bit in the third byte, etc..

For example if the response was 686Bxx62492355 and the "55" was the data we wanted, RXD would be 3008. The "30" is hex for 48 bits or the start of the number 6 byte into the response (the first byte is 00). The "08" means the response is 8 bits in length.

CAN Vehicles

CAN vehicles treat the response similar to the non-CAN vehicles. Internally, the ScanGaugeII treats the Identifier as a 3 byte value, even though in TXD it was only 2 bytes. By doing this, the RXD is the same for both CAN and non-CAN responses.

MTH

MTH tells the ScanGaugeII how to scale and offset the data for the correct display. There are 3 fields in MTH.

Multiplier	Divider	Add/Subtract
0000	0000	0000

The Multiplier is a 2 byte value to multiply the data by. The resulting data must not be larger than 4 bytes and the upper bit must be 0 if the resulting value is not a negative value.

The Divider is a 2 byte value to divide the results from the multiply by.

The Add/Subtract adds the 2 byte value to the results from the multiply and divide. This is a signed value. If the upper bit is set, the value is the two's compliment negative value. If the result has the highest bit set, the response will assume it is a two's compliment negative value and will show it as a negative value. Some care is required to avoid unintentionally setting or clearing this bit through an add/subtract operation and changing the sign of the displayed value. This can be done by scaling the value properly using the multiply and divide before doing the add/subtract.

NAME

This is a 3 character field you use to set the name to display in the gauge mode. The characters include the entire ASCII character set and the extended characters in the display of the ScanGaugeII.

Note: You can use the space character to reduce the number of characters in the name. The space is reached at the beginning of the punctuation. Getting to it requires you to decrement significantly from the capital letters. This can be done by holding down the decrement button.