



USER MANUAL

Digital Gear Indicator

Document version: 5.0

Firmware version: 4.0 or later

Published on: 11 May 2026



Contents

1. Device description	3
2. Specification	3
3. Pinout	5
4. PC connection	5
5. Configuration	5
5.1. RPM settings.....	6
5.2. Gear settings.....	6
5.3. Shift light.....	9
5.4. Brightness.....	10
5.5. CAN Stream.....	10
6. Document history	11

1. Device description

The Digital Gear Indicator is a device designed to display gear information and provide a shift light to the driver. It can read gear, RPM, brightness, and shift light data either from the CAN bus or through its built-in analog and digital inputs. Additionally, the device transmits gear and RPM information via CAN bus using the Export ID.

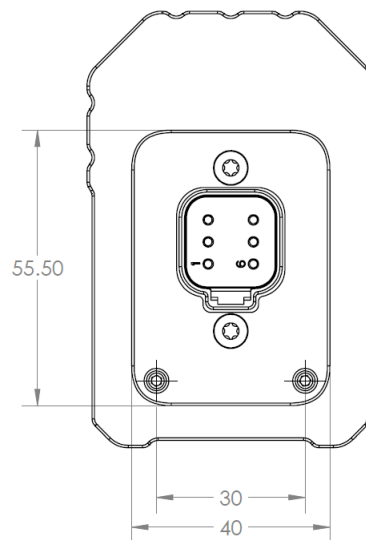
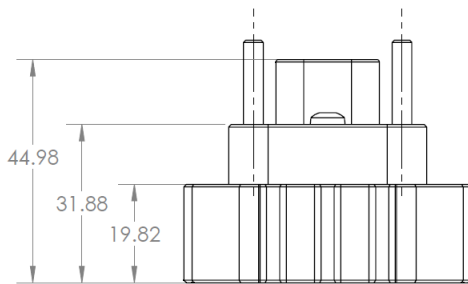
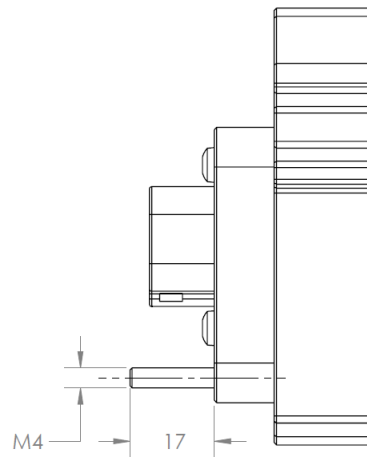
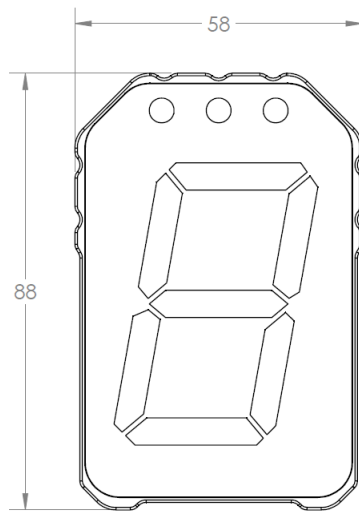
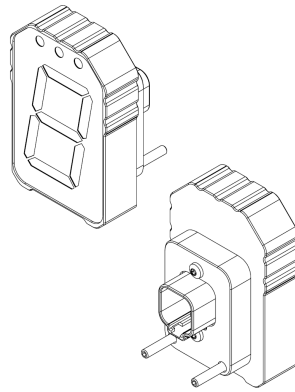
Key Features

- Clear and precise display of the current gear for the driver.
- Integrated shift light to signal optimal gear change timing.
- Reads data from the CAN bus or from built-in analog and digital inputs.
- Transmits gear and RPM information using Export ID.
- Adjustable brightness for optimal visibility in various lighting conditions.

2. Specification

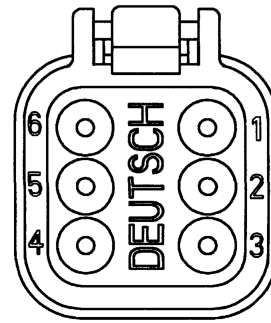
Specification	
Voltage range	6V - 24 V (12 V automotive installations)
Analog input	0-5 V
Digital input	TTL 0-5 V (VR sensor tolerant up to 100 V), 0.1 Hz-50 kHz, rising or falling edge
Dimensions	88 x 58 x 45 mm
Weight	160 g
Connector	Deutsch DTM06
PC communication	Using USB to CAN interface (Ecumaster USBtoCAN, Peak, Kvaser)
CAN bus bitrate	1 Mbps (default), 500 kbps, 250 kbps, 125 kbps
CAN termination	None

All dimensions in mm



3. Pinout

Pin	Description
1	+12 V
2	GND
3	Analog input
4	Frequency input
5	CAN low
6	CAN high



4. PC connection

Communication with the PC software is done via CAN bus, which requires a special interface to convert CAN bus communication to USB 2.0. This interface is an independent device and must be purchased separately.

Supported interfaces:

- Ecumaster USBtoCAN (driver and manual: <https://www.ecumaster.com/products/usb-to-can/>)
- PEAK-System
- Kvaser

5. Configuration

Ecumaster Light Client is the configuration software for many Ecumaster products and also serves as a tool for CAN bus monitoring.

Light Client can be used to check available devices on the CAN bus, display channel values, change CAN bus bitrate, and modify device specific properties. CAN bus monitoring is made possible through a list of frames grouped by ID, the ability to save traffic trace files, and the option to send custom messages on the CAN bus.

The software and manual can be downloaded from: <https://www.ecumaster.com/products/light-client/>

5.1. RPM settings

When the *RPM source* is CAN

Parameter	Description
RPM source	CAN
RPM input CAN ID	CAN-bus frame defined via ID
Byte position	Value must be between 0 and 7
Num bytes	1 byte / 2 bytes (8 bit / 16 bit) - number width
Endian (Only for 2 bytes)	<i>big endian / little endian</i> - i.e. the "sequence" of bytes for 16-digit numbers. It shows how a number stored in two consecutive bytes is to be interpreted. E.g. numbers 0x12, 0x34 can be interpreted as 0x1234 for the <i>big endian</i> or 0x3412 for the <i>little endian</i>
Divider	For scaling the value
Multiplier	For scaling the value

When the *RPM source* is "CAN-OBD" default settings for OBD II are used.

When the *RPM source* is *FREQ*

Parameter	Description
RPM source	<i>FREQ</i> – RPM is read from digital frequency input
Signal edge	<i>Falling / Rising</i> - Choose signal edge that has faster passing through 2.5V threshold
Teeth per 720° (one full engine cycle – 2 revolutions)	Enter here number of digital signals per engine cycle. If you have trigger wheel with missing teeth, enter number EXCLUDING missing teeth. For example for 60 - 2 wheel enter " $2 \times (60 - 2) = 116$ "

5.2. Gear settings

When the *Gear source* is CAN

Parameter	Description
Gear source	CAN
Gear input CAN ID	CAN-bus frame defined via ID

Parameter	Description
Byte position	Value must be between 0 and 7
Num bytes	1 byte / 2 bytes
Endian (Only for 2 bytes)	<i>big endian / little endian</i> - i.e. the “sequence” of bytes for 16-digit numbers. It shows how a number stored in two consecutive bytes is to be interpreted. E.g. numbers 0x12, 0x34 can be interpreted as 0x1234 for the <i>big endian</i> or 0x3412 for the <i>little endian</i>
Type	<i>signed / unsigned</i> - signed is a number with a sign (it can receive positive and negative values, as well as zero). An example of such value is the value from the cooling liquid temperature sensor. Unsigned – numbers zero and above. For example engine speed (RPM)
Extract bitfield	take only a part of an 8- or 16-bit number, (for example, to check the setting of a bit of a 0x80 mask the following settings should be used: Bit count: 1, Bit position: 7)
Bit count	Value must be between 1 and 16
Bit position	Value must be between 0 and 15
Offset	Value added to the raw CAN signal to convert it into a real-world value.
Park value	In decimal notation
Reverse value	In decimal notation
Neutral value	In decimal notation

Example 1

Below is the gear data. We will need to read 1 byte, signed. The signal are on CAN ID 0x400.

Reverse – FF (Reverse value in decimal is -1)

Neutral – 00

1st Gear – 01

2nd Gear – 02

3rd Gear – 03

4th Gear – 04

5th Gear – 05

6th Gear – 06

7th Gear – 07

Gear settings:	
Gear source	CAN
Gear input CAN ID	0x400 Standard
Byte position	0
Num bytes	1
Type	Signed
Extract bitfield	<input type="checkbox"/>
Offset	0
Park value	-2
Reverse value	-1
Neutral value	0

Example 2

Below is the gear data. We will need to read 1 byte, unsigned, extract 4 bits, starting from bit 0-th. The signal are on CAN ID 0x400.

- Reverse – 2F (0010 1111)
- Neutral – 00 (0000 0000)
- 1st Gear – 11 (0001 0001)
- 2nd Gear – 12 (0001 0010)
- 3rd Gear – 13 (0001 0011)
- 4th Gear – 14 (0001 0100)
- 5th Gear – 15 (0001 0101)
- 6th Gear – 16 (0001 0110)
- 7th Gear – 17 (0001 0111)

Gear settings:	
Gear source	CAN
Gear input CAN ID	0x400 Standard
Byte position	0
Num bytes	1
Type	Unsigned
Extract bitfield	<input checked="" type="checkbox"/>
Bit count	4
Bit position	0
Offset	0
Park value	10
Reverse value	15
Neutral value	0

Reverse value will be 0x0F (15 – in decimal notation).

Example 3

Below is the Ford F150 data that was read with the Light Client. All The signals are on Can ID 0x171.

- Park – A5 00 A0 00 00 00 00 00
- Reverse – A5 20 A0 00 00 00 00 00
- Neutral – 15 40 A0 00 00 00 00 00
- 1st Gear – 15 80 A0 00 00 00 00 00
- 2nd Gear – 25 80 A0 00 00 00 00 00
- 3rd Gear – 35 80 A0 00 00 00 00 00
- 4th Gear – 45 80 A0 00 00 00 00 00
- 5th Gear – 55 80 A0 00 00 00 00 00
- 6th Gear – 65 80 A0 00 00 00 00 00
- 7th Gear – 75 80 A0 00 00 00 00 00
- 8th Gear – 85 80 A0 00 00 00 00 00
- 9th Gear – 95 80 A0 00 00 00 00 00

First nibble from the left is gear number. So some bit extraction is needed. We will need to read 2 byte little endian (16 bits). Extract 11 bits, starting from bit 4-th.

- 00 A5 – for park (0000 0000 1010 0101)
- 20 A5 – for reverse (0010 0000 1010 0101)
- 40 15 – for neutral (0100 0000 0001 0101)
- 80 15 – for 1st Gear (1000 0000 0001 0101)
- 80 25 – for 2nd Gear (1000 0000 0010 0101)
- 80 35 – for 3rd Gear (1000 0000 0011 0101)
- 80 45 – for 4th Gear (1000 0000 0100 0101)
- 80 55 – for 5th Gear (1000 0000 0101 0101)
- 80 65 – for 6th Gear (1000 0000 0110 0101)
- 80 75 – for 7th Gear (1000 0000 0111 0101)
- 80 85 – for 8th Gear (1000 0000 1000 0101)
- 80 95 – for 9th Gear (1000 0000 1001 0101)

Gear settings:	
Gear source	CAN
Gear input CAN ID	0x171 Standard
Byte position	0
Num bytes	2
Endian	Little endian
Type	Unsigned
Extract bitfield	<input checked="" type="checkbox"/>
Bit count	11
Bit position	4
Offset	0
Park value	10
Reverse value	522
Neutral value	1025

Park value will be 0x00A (10 – in decimal notation), reverse value 0x20A (522) and neutral value 0x401 (1025).

When the *Gear source* is AIN

Parameter	Description
Gear source	AIN - analog input – gear sensor calibration with 9 position Voltage table. Current voltage is visible in Channels log.
Offset	Value added to the raw CAN signal to convert it into a real-world value.
Reverse	This section contains voltage values for gear reading from analog input. Gear with voltage closest to analog input voltage is set.
Neutral	
# (1-7)	

When the *Gear source* is VAG DSG (ECU CAN) default settings are used.

5.3. Shift light

Parameter	Description
LED # (1-3) color	color on # shift light led
LED # (1-3) RPM	LEDs are lit based on RPM thresholds
ALL RED RPM	threshold for overriding all LEDs to red color
BLINK RPM	threshold for all LED blinking

5.4. Brightness

Parameter	Description
LED	Brightness settings for Shift Light LEDs. Values from 0 - 100
Gear	Brightness settings for Segment display. Values from 0 - 100

Temperature reading is only for overheat protection. Precision is +/- 10 °C



Important:

Device has brightness limit in case of overheating. Brightness is limited linearly from 100% at 40°C to 0% at 80°C.

5.5. CAN Stream

Channels are the data values sent over CAN bus. Channels are sent as raw values, which means that obtaining values with correct units requires some calculations.

$$Value[unit] = \frac{Value[raw] * Multiplier}{Divider} + Offset$$

The table below describes how each channel is positioned inside CAN frame and how to obtain the correct value.

The default CAN bus bitrate of the device is **1 Mbps**. The format used is **big-endian**.

Ecumaster CAN profile description

Byte	Bit	Channel	Data type	Range	Multiplier /Divider	Factor	Offset	Unit
Export CAN ID+0 (default: 0x6A4)								
0..1		Ain	16-bit U	0 – 20000	1/1	1	0	mV
2..3		Engine speed	16-bit U	0 – 20000	1/1	1	0	RPM
4..5		Temperature	16-bit S	-50.0 – 200.0	1/10	0.1	0	°C
6		Gear number	8-bit S	-128 – 127	1/1	1	0	
7		Heartbeat	8-bit U	0 – 255	1/1	1	0	

6. Document history

Version	Date	Changes
0.2	2019.06.14	Initial revision
4.0	2023.01.13	Clarified that the device only works in 12 V installations Updated to version 4.0
5.0	2026.05.11	Document layout updated to follow the Ecumaster standard format The structure and text have been refined and improved for better readability and clarity