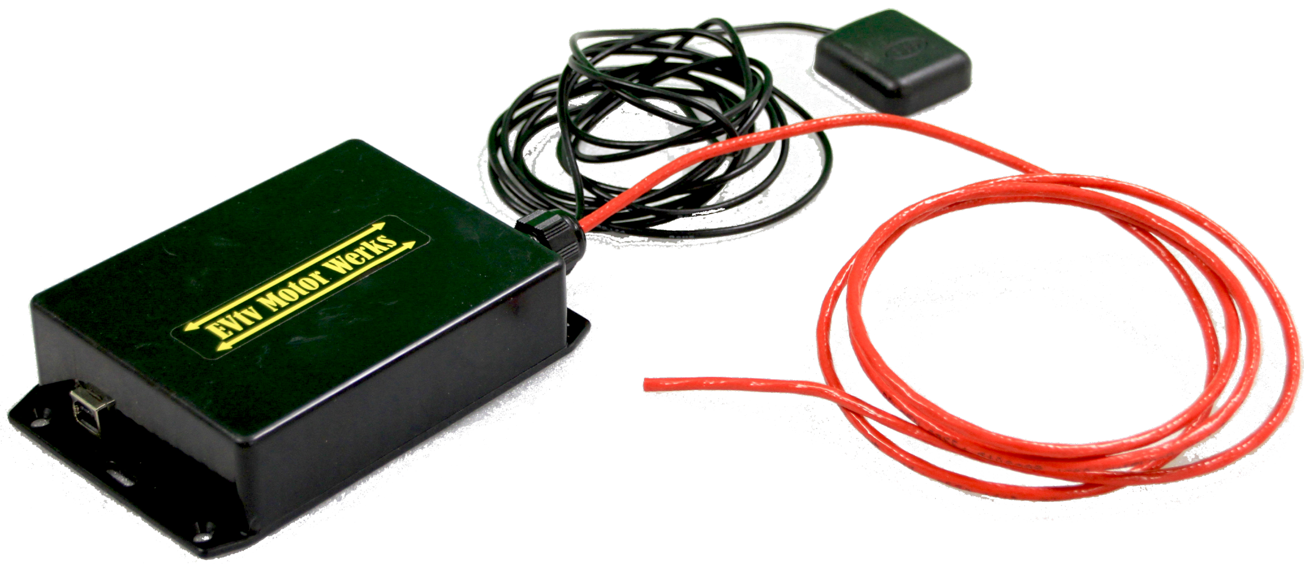
EVTV CAN-GPS

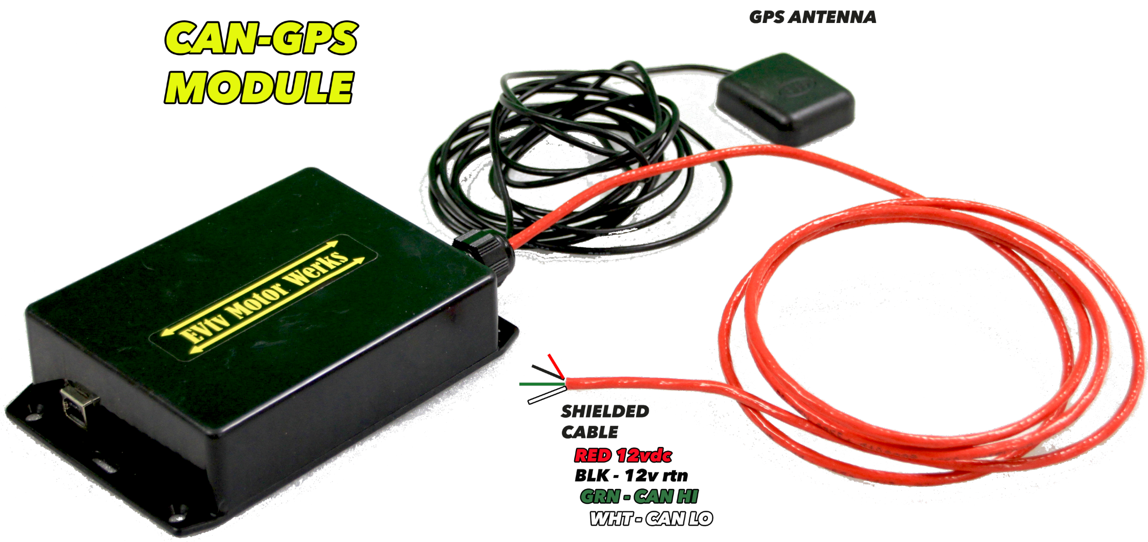


Controller Area Network

Global Positioning System

Receiver and Real Time Clock

INTRODUCTION



The EVTV Controller Area Network Global Positioning System Receiver (CAN-GPS) is a simple module that can receive GPS position data from the GPS satellite network and make it available over a Controller Area Network (CAN) bus in standard CAN 2.0B data frames.

The device consists of the EVTVDue, an Arduino Due compatible microcontroller with a Controller Area Network (CAN) port. It also features a GPS shield and antenna using the RoyalTek REB-5216 GPS module.
This is a 52 channel CSR SiRFstarV 5e CSRG0530 chipset. It is capable of receiving both GPS and GLONASS satellite transmissions.

It has an autonomous acquisition sensitivity of -146dBm and a tracking sensitivity of -165dBM giving it excellent sensitivity for foliage or urban canyon environments. Accuracy is within about 10 meters and it provides updates once per second. It runs on 3.3v at about 35 mA normally.

It provides the following data via three specific CAN message IDs.

- Current Speed in U.S. statute miles per hour.
- Direction of Travel - Course in degrees.
- Odometer - miles traveled since last reset.
- Current latitude in 32 bit precision
- Current longitude in 32 bit precision
- Universal Coordinated Time real time mark

By default, this information is provided in CAN data frame message ID **0x700** and **0x701** using standard 11-bit CAN message addresses at a data rate of 250 kbps (default). Messages are sent once per second.

Additionally, CAN message **0x702** provides time of day and calendar data even when GPS transmissions are incomplete or invalid.

However, you can modify this to ANY message address you like, in either 11-bit or 29-bit addresses, and at any data rate. You do this by sending a control message with message ID **0x711**, or by USB Serial Port connection and an ASCII terminal.

This system is intended for automotive use and is powered by the standard 12v vehicle power system.

The module consists of a black plastic enclosure with an external GPS antenna and four colored wires:

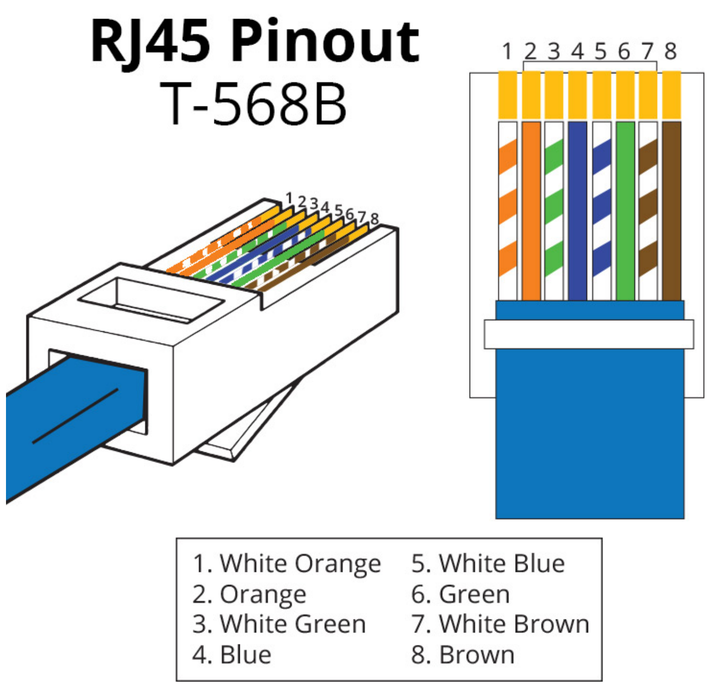
RED	**12v vehicle power**
BLK	**12v return (frame ground)**
GRN	**CAN HI**
WHT	**CAN LO**

An optional RJ-45 T-568B CAT5 cable option is available.

The output signals:

Pin 1 orange/white = 12vdc in
Pin 4 blue = CANHI
Pin 5 blue/white = CANLO
Pin 8 brown = GND

12v power starts the module.
CAN is received on the pin 4/5 pair



MESSAGE DATA FORMATS

The CAN-GPS module will issue a Controller Area Network version 2.0B compliant CAN message about once per second. This can be adjusted to a greater frequency. However, since the GPS updates at a 1 Hz rate, it would contain no more information and so the default is to issue two messages - once each second.

These messages are defined below. Message values are presented in hexadecimal (base 16) format.

0x700 08   61 6E BB 1E 43 14 24 35

Message 0x700 contains 8 data bytes, conventionally numbered 0 through 7.

BYTE 0-1. This is a 16bit unsigned integer representing our course direction * 100. This is in standard Intel Least Significant Bits/Most Significant Bits (LSB/MSB) order. 0x616E then is hexadecimal 6E1 or decimal value of 28257. If we divide this by 100 we get a course direction of 282.57 degrees.

BYTE 2-3. This value represents our odometer or the total distance traveled since the module was last powered on. This is a 16-bit unsigned integer representing statute miles *100 in LSB/MSB format.

1E BB represents a decimal value of 7867. If we divide this by 100 we get a trip odometer value of 78.67 U.S. statute miles.

Note this value can be reset to zero at any time with a 0x711 control message.

BYTE 4 Speed in whole statute miles per hour. 0x43 represents 67 mph.

BYTE 5. This is a simple 8-bit integer representing the HOUR of the current UTC Universal Coordinated Time (Greenwich Mean Time). In this case hour 14.

BYTE 6 This is a simple unsigned 8-bit integer representing the MINUTE UTC. In this case minute 24.

BYTE 7 This is a simple unsigned 8-bit integer representing the SECOND of UTC time. In this case 35 seconds.

NOTE: GPS signals are updated at the rate of 1 per second. The values are then encoded into CAN messages and transmitted on the CAN bus at a configurable rate. The GPS time can be used to set a real time clock on the system to an accuracy of about a second. So while GPS transmits sub second values, there is not much point in putting them on the CAN bus as the transit times preclude the usefulness of such accuracy.

0x701 08 59 F6 5D 02 BB 14 24 35

Message 0x701 contains 8 data bytes, conventionally numbered 0 through 7. These bytes represent just two values - latitude and longitude.

This is done in two four-byte, signed 32-bit integers. Positive values represent Western Hemisphere in the case of longitude and Northern Hemisphere in the case of latitude. Similarly, negative values would be for Eastern Hemisphere longitudes and Southern Hemisphere latitudes.

These 32-bit values are “little endian” meaning that the first byte contains the least significant bits LSB and the last byte contains the most significant bits MSB.

This allows us quite a bit of precision in our latitude and longitude calculations - up to 6 decimal points in fact.

But as integers only hold whole numbers, we transmit them as very large numbers. To retrieve these values, we must also perform a division by a value of 1,000,000 to properly place the decimal point.

So for a lattitude of 39.712345, we store this as the value 39,712,345. This corresponds to a hexadecimal value of 02 5D F6 59. But as we can see in our message, the first four bytes contain that value REVERSED in little endian format. This allows you to easily load those bytes into a 32-bit integer variable

And again, to get true latitude we must divide by 1,000,000 to get back to our 39.712345 value.
Longitude is handled precisely the same way for East of Greenwich England (negative) or West of Greenwich (positive).

0x702 06 11 0C 1B 0F 2A 33

Message **0x702** contains just 6 data bytes, and it is somewhat interesting in that it updates even when the gps NEMA messages are invalid or incomplete.

Many applications don't really need gps position data, but knowing the time of day or date can be quite convenient for some tasks involving scheduling. Unfortunately, very few microcontrollers feature a Real Time Clock. The CAN GPS module will issue this CAN command to provide time and date information.

Initial GPS NEMA messages from the module are invalid and incomplete for some period of minutes after coming online. But a very basic message is sent and it contains time of day in Universal Coordinated Time (UTC) format as well as year, month and day.

CANGPS will use that info to send this message even when no valid navigational data is available.

Byte 0 = last two digits of year in hexadecimal format. Ergo **0x11** would be the equivalent of decimal 17 or the year 2017.

Byte 1 = month. **0x0C** would be month 12 or December.

Byte 2 = day with **0x1B** indicating 27.

Byte 3= Hour. **0x0F** would indicate 15 or 3:00PM universal coordinated time (UTC). If your time zone is, for example, the Central Time Zone in the U.S. and there is no daily savings time in effect, you are CST or UTC-6 which would be 9:00AM local time.

Byte 4: Minute. **0x2A** hexadecimal would indicate a minute value of 42.

Byte 5: Second. Hexadecimal **0x33** indicates second 51.

CONTROL MESSAGE FORMAT

0x711 08 00 06 00 00 B9 1E FA 00

Message ID 0x711 is always recognized by the GPS-CAN as a control message. This message contains 8 data bytes, conventionally numbered 0 through 7.

BYTE 0-3 New data message ID. This is a 32-bit integer LSB first (little endian). If zero, this 32-bit value causes NO change.

Any value above zero will be taken as the NEW base address for the data messages put out by the CAN-GPS.

The default address is an 11-bit message ID 0x700. A second message is also provided one higher at 0x701.

The example shown above would change those message IDs to 0x600 and 0x601.

Note that this value is fully capable of transmitting larger 29-bit addresses as well.

BYTE 4-5. This is a 16bit unsigned integer LSB/MSB representing our distance traveled or odometer. If this value

is set to zero, it will reset our odometer to zero statute miles.

However, it can also be used to set the odometer to any value to two decimal places of precision.

For example, to set odometer to 78.65 statue miles, multiply the value *100 to create the whole number 7865 and store in this two byte segment. In hexadecimal, this would appear as 1E B9

The maximum value would then be 65534/100 or 655.34 miles.

The value 65535 decimal or 0xFFFF hex is reserved to indicate NO change to the odometer. For example, if you are changing data rate or message IDs but do not wish to change the odometer setting, set this to 0xFFFF.

BYTE 6-7. This is a 16-bit unsigned integer LSB/MSB indicating data rate. If no change, set to 0x0000.

To change the data rate of the transmitted date message IDs, simply set this to a whole number representing kbps.

For example, to change the data rate from the default 500 kbps to 250 kbps, set this value to 250 decimal or 0x FA 00.

Changes to any or all of these values will take effect immediately on receipt of the 0x711 message by the CAN-GPS unit.

Message ID 0x711 is very low priority on the CAN message collision arbitration scheme. You may need to repeat the message until you observe the change in odometer, base address, or data rate.

Note that the address of the control message is reserved and cannot be changed by CAN.

SERIAL USB PORT

The CANGPS module features a mini-B style USB serial port and provides data using American Standard Code for Information Interchange (ASCII) text. There are also a few items of configuration you can make.

To connect to the port, use any ASCII terminal program setting the serial port values to 115,200 bps, 8 data bits, no parity, 1 stop bit (8N1).

Once the GPS obtains a fix, it will issue an ASCII string each time the GPS updates its position over the other serial port.

```
Current position:37.297554N 89.519630W Speed:0.35 mph Course:30.32 degrees Odometer:0.06 miles
Date:12/27/17 Time:15:37:24 UTC
```

Enter an “h” for help or “?” to see the menu below:

```
===========  CAN-GPS  Version 3.01 ==============
************ List of Available Commands ************

  ? or h  - Print this menu
  d - toggles debug DISPLAY FRAMES to print CAN data traffic
  i - set interval in ms between CAN frames i550
  r - set data rate in kbps ie r500 for 500,000 bps or 500 kbps
  z - zero distance traveled odometer
**************************************************************
==============================================================
```

By entering a D or "d" with a carriage return or line feed, you will toggle the DEBUG mode on or off. When on, the full data on CAN frames transmitted, the NEMA string received from the GPS module, etc. will be displayed in gory detail for troubleshooting purposes.

CAN data is actually transmitted on receipt of NEMA messages and update of the gps object, but you can change the screen refresh rate by entering I followed by the number of milliseconds between refresh. Ie I200.

One important characteristic is the speed of the CAN port operations, which can be set with the R command as in R500 for 500 kbps or R250 for 250 kbps data rates.
Finally, you can zero the odometer with a Z command.

Most of these values can likewise be set remotely by sending the 0x711 CAN control message described earlier.