OBDII Odometer Reference Manual

Table of Contents

1 Revision History

Version	Date	Changes
1.1	26/11/2014	Added firmware changelog, section 2 Added hardware changelog, section 3 Updated power source, section 7.2 Updated standalone testing, section 8.5 Updated odometer flags, section 9.7.1.1 Added OBDII protocol, section 9.7.1.2
1.0	29/07/2014	Initial Release

2 Firmware Changelog

Version	Date	Changes
1.1	26/11/2014	System now outputs active protocol in previously reserved flags data System now outputs zero velocity when it cannot connect to an ECU (assumption that car is turned off and stationary) Bug where system would overload ECU on some older US cars has now been resolved
1.0	29/07/2014	Initial Release

3 Hardware Changelog

Version	Date	Changes
1.0	29/07/2014	Initial Release

4 Foundation Knowledge

This chapter is a learning reference that briefly covers knowledge essential to understanding the OBDII Odometer and the following chapters. It explains the concepts in simple terms so that people unfamiliar with the technology may understand it.

4.1 OBDII

OBDII (On-board Diagnostics) is a standardised electrical data interface installed in most modern road vehicles that allows external systems to read information from the vehicle computer in real time. It's original purpose was to assist in diagnosing faults with the vehicle's engine management system, however it has since expanded to allow for reading data from other vehicle control systems such as the transmission and ABS (Anti-lock braking system) which can provide information on the vehicle's speed.

4.2 Odometer

An odometer is a device that measures the distance a vehicle has travelled. This information is very useful for navigation systems.

5 Introduction

Advanced Navigation's OBDII Odometer is a smart interface cable that communicates with a vehicle's computer to determine the current vehicle speed. It outputs real time vehicle speed information over an RS232 serial data interface.

The speed data that the OBDII Odometer outputs is extremely useful for vehicle navigation systems such as Advanced Navigation's Spatial series of GPS/INS products. The OBDII Odometer allows a Spatial GPS/INS to maintain significantly higher position, velocity and orientation accuracy when GPS is not available with position error growth as low as 0.05% distance travelled.

6 Specifications

6.1 Mechanical Drawings

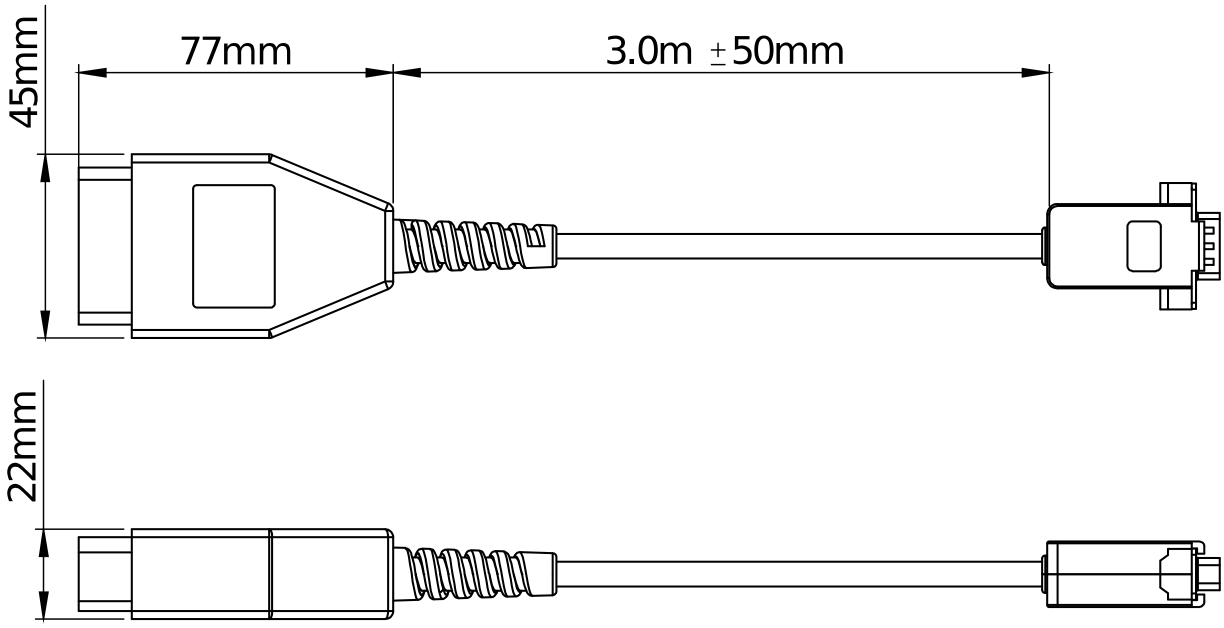

Illustration 1: Mechanical drawings of OBDII Odometer

6.2 OBDII Specifications

Parameter	Value
Standardised Protocols Supported	ISO 9141-2 ISO 11898 ISO 14230-4 ISO 15765 ISO 15765-4 SAE J1850 VPW SAE J1850 PWM

Table 1: OBDII specifications

6.3 Communication Specifications

Parameter	Value
Interface	RS232
Interface Isolation	Optically Isolated
Speed	115200
Protocol	AN Packet Protocol
Output Data Rate	10 Hz

Table 2: Communication specifications

6.4 Hardware Specifications

Parameter	Value
Operating Voltage	9 to 16 volts
Input Protection	Surge Over-voltage and under-voltage Brown out Reverse polarity
Power Source	OBDII Connector (vehicle power)
Power Consumption	60 mA @ 12 V (typical)
Operating Temperature	-40 °C to 85 °C
Environmental Protection	IP54
Weight	202 grams

Table 3: Hardware specifications

6.5 Electrical Specifications

Parameter	Minimum	Typical	Maximum
RS232			
Tx Voltage Low		-5.7 V	-5 V
Tx Voltage High	5 V	6.2 V	
Tx Short Circuit Current		±35 mA	±70 mA
Rx Threshold Low	0.8 V	1.3 V	
Rx Threshold High		1.7 V	2.5 V

Table 4: Electrical specifications

6.6 Connector Pin-out

The RS232 data signal connections are made through an industry standard male DB9 connector with 3 metres of shielded cable.

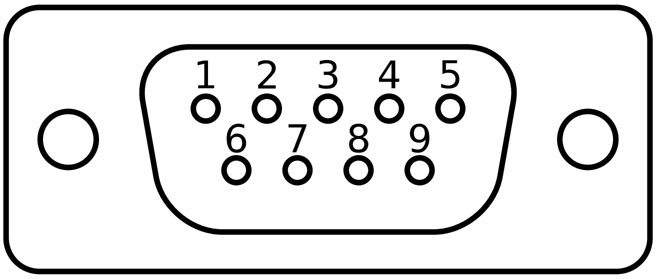

Illustration 2: DB9 connector pin-out for OBDII Odometer

Pin	Colour	Function
2	Blue	RS232 Receive
3	Green	RS232 Transmit
5	White	Ground

Table 5: Pin allocation table

7 Installation

7.1 Connection to vehicle

To install the OBDII Odometer into a vehicle, it must be equipped with an OBDII interface. Almost all road cars worldwide built since the mid 1990s are equipped with OBDII. In the United States it has been mandatory for all new vehicles sold to be equipped with OBDII since 1996.

The OBDII standard specifies a 16 pin J1962 connector that is the same across all vehicles. The OBDII connector is required to be located within 0.6 metres of the steering wheel and is typically found in the driver side foot well. Please see Illustration 3 and Illustration 4.

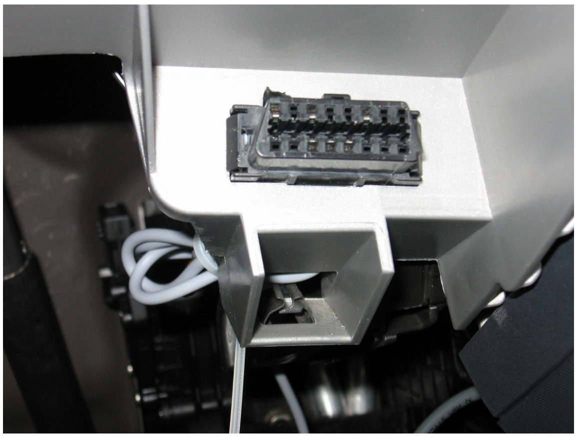

Illustration 3: Vehicle OBDII connector

Illustration 4: OBDII Odometer plugged in

To install the OBDII Odometer, plug it into the OBDII connector. It does not matter whether the ignition is on or off when the OBDII Odometer is plugged in. Due to the OBDII connectors proximity to the pedals in many cars, Advanced Navigation strongly recommends that the OBDII Odometer cable is secured using cable ties to avoid it getting tangled up in the pedals and causing a safety hazard.

7.2 Power Source

The OBDII Odometer is powered by the vehicle's power supply over the OBDII connector and requires no external source of power. The OBDII Odometer is always powered when it is plugged in, irrespective of whether the car's ignition is switched on or off. It is important to be aware that the vehicle's battery will discharge if the OBDII Odometer is left plugged in when the engine is not running. If the vehicle is left with the OBDII Odometer plugged in and the engine not running, it will discharge the battery to the extent that the engine will not be able to start after approximately 21 days. For this reason, Advanced Navigation recommends unplugging the OBDII Odometer when the vehicle is not in use. For permanent installations it is recommended to have an auto electrician modify the vehicle's OBDII port so that the OBDII power is switched off with the ignition.

The OBDII Odometer's electrical data signals are optically isolated from the vehicle's power supply to avoid ground loops and interference with other equipment.

8 Operation

8.1 Speed Data Source

There are two possible sources of speed in the vehicle's computer, the transmission speed and the ABS (Anti-lock Braking System) speed. The ABS speed is of a higher precision, however there is no standard for reading data from a vehicles ABS system. The OBDII Odometer contains an internal database for reading ABS data from some of the major vehicle manufacturers systems. The OBDII Odometer automatically selects the speed source. If the vehicle is contained in the OBDII Odometer's ABS database then the system will use ABS speed, if not it falls back to transmission speed which is less precise than ABS speed but still very useful and is standardised across all OBDII vehicles. Transmission speed and ABS speed have different precision but both have roughly the same accuracy which varies from vehicle to vehicle.

Speed Source	Precision
Transmission	0.28 metres/second
ABS	0.025 metres/second

Table 6: Typical speed precision over OBDII

8.2 Reverse Detection

The OBDII Odometer supports reversing detection on several of the major automotive manufacturers. The OBDII standard does not specify data for determining if the vehicle is reversing, however some automotive manufacturers have implemented their own non-standard methods for determining if the vehicle is reversing over OBDII. If reversing detection is supported with a vehicle, the OBDII Odometer speed will show as negative when travelling in reverse. If reversing detection is not supported on a vehicle the speed will show as positive whether travelling in a forward or reverse direction.

8.3 Delay Estimation

The OBDII Odometer is able to estimate delay from motion to output of the real time speed data. This works on most vehicles. The OBDII Odometer also contains a database of typical delays for a range of vehicles which it will use if it cannot estimate delay. If the delay cannot be estimated and there is not a typical value contained in it's database for the vehicle then it will output a fixed delay value of 0.3 seconds.

8.4 Use with Advanced Navigation Spatial series GPS/INS

All Advanced Navigation's Spatial series of GPS/INS products support the OBDII Odometer. To connect and set it up please follow the instructions below.

8.4.1 Connection

The OBDII Odometer can be connected directly to a Spatial Dual or Spatial FOG with the standard interface breakout cable included in the evaluation kit. To use the OBDII Odometer with Spatial an interface breakout cable needs to be purchased which is not included in the standard evaluation kit. The OBDII Odometer should be plugged into the Auxiliary RS232 connector of the interface breakout cable, see Illustration 5.

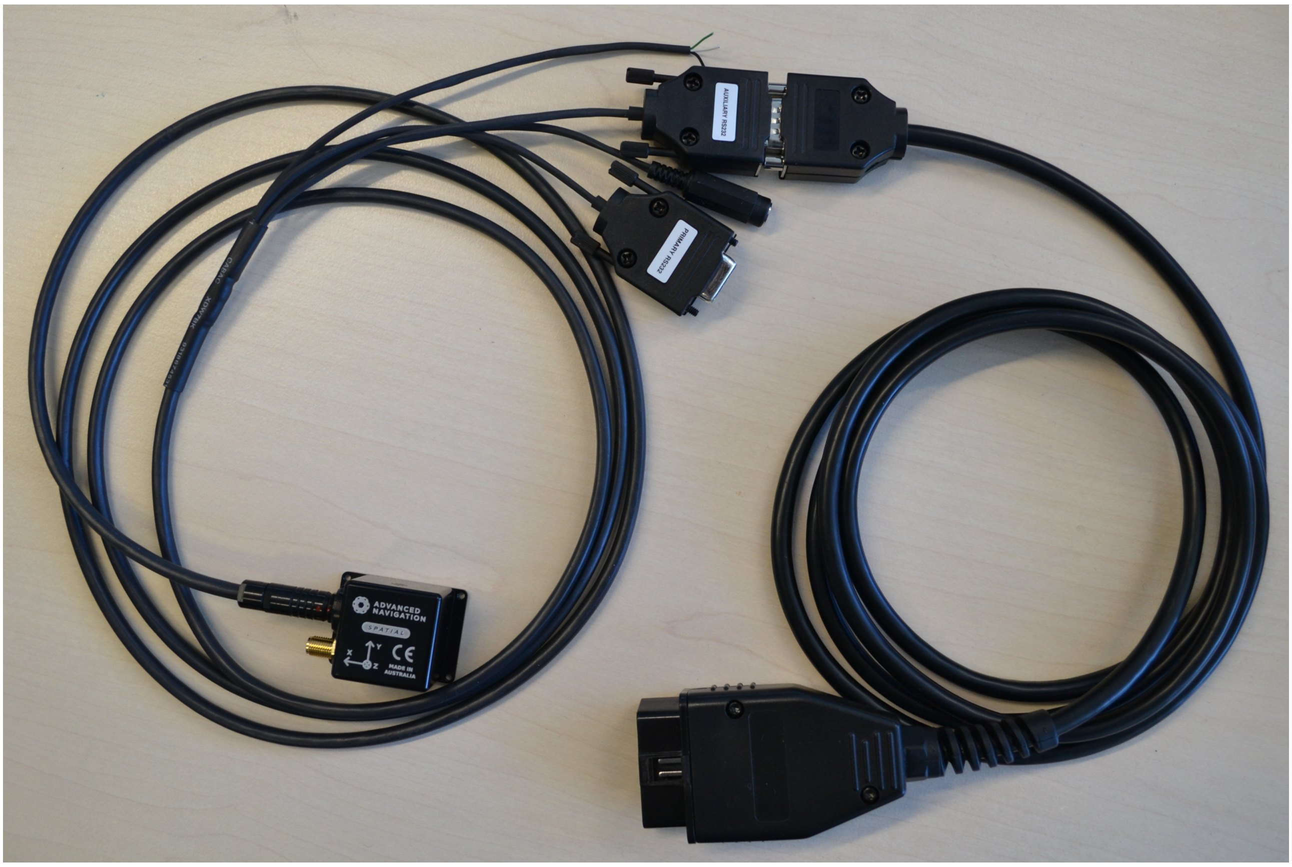

Illustration 5: OBDII Odometer connected to Spatial

8.4.2 Software Configuration

Use the following steps to setup a Spatial series device for use with the OBDII Odometer.

1. Open Spatial Manager and connect to the Spatial device. See Step 1 in Illustration 6 below.
2. In the configuration menu, open the GPIOs dialogue, set the Auxiliary RS232 Function to ANPP Input and save. See Step 2 in Illustration 6 below.
3. In the configuration menu, open the Baud Rates dialogue, set the Auxiliary Port Baud Rate to 115200 and save. See Step 3 in Illustration 6 below.
4. To verify that the system is functioning correctly, in the view menu, open the Status dialogue. Turn on the vehicle ignition and after a few seconds the External Velocity box should become ticked. See Step 4 in Illustration 6 below.

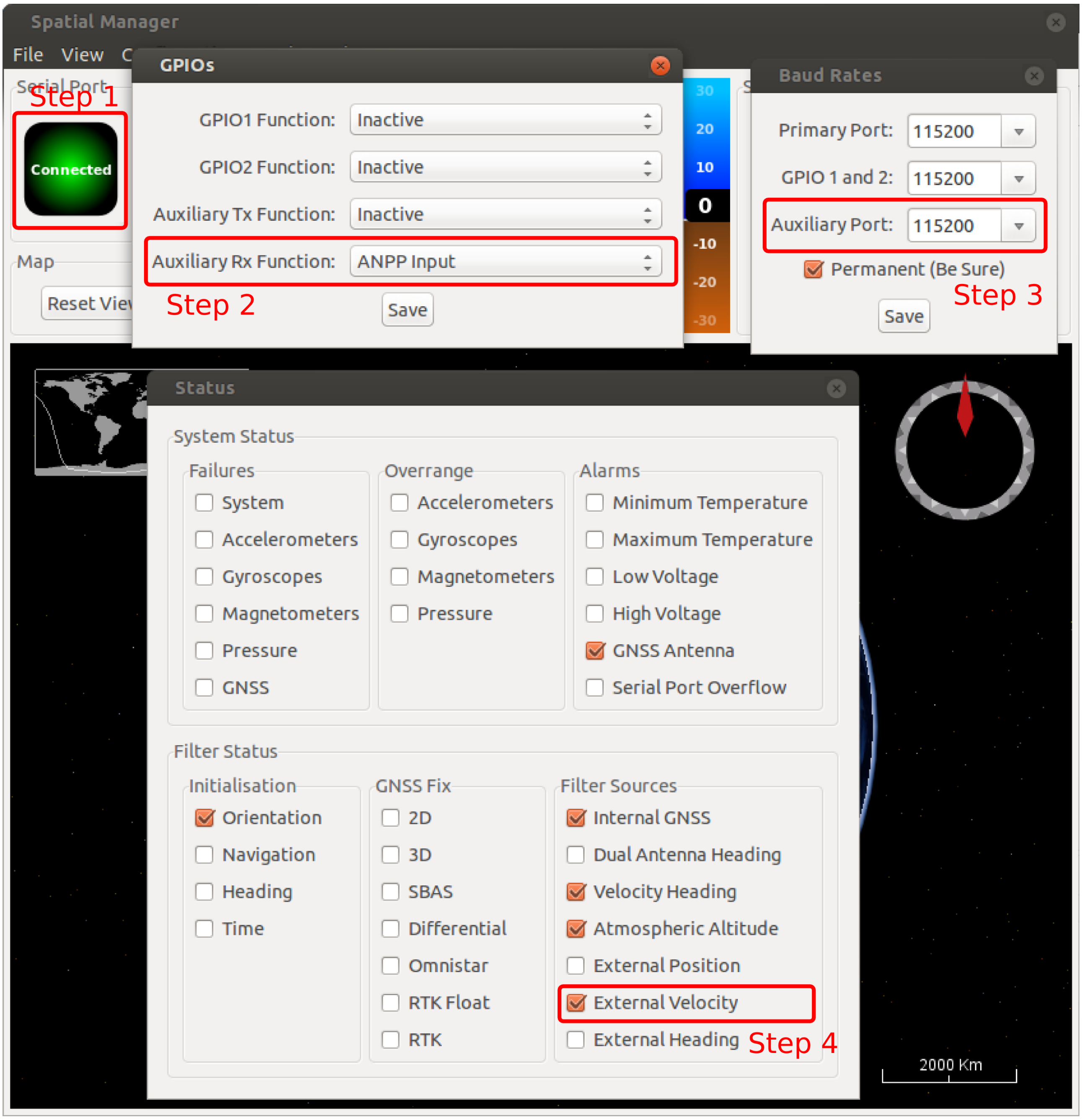

Illustration 6: Spatial Manager configuration steps

It is recommended to update your Spatial device to the latest firmware for best performance with the OBDII Odometer.

8.5 Standalone Testing

Advanced Navigation also supplies a basic software utility called OBDII Odometer Viewer that can be used to check the output from the OBDII Odometer. OBDII Odometer Viewer can be downloaded from the software section of the OBDII Odometer page on the Advanced Navigation website. Java software must be installed to run this utility. Java can be downloaded from the Java website at this address: http://www.java.com. Illustration 7 shows a screenshot of the OBDII Odometer Viewer utility.

The OBDII Odometer can be connected to a computer using a null modem RS232 cable and a USB to RS232 adapter. These are not supplied as standard with the OBDII Odometer.

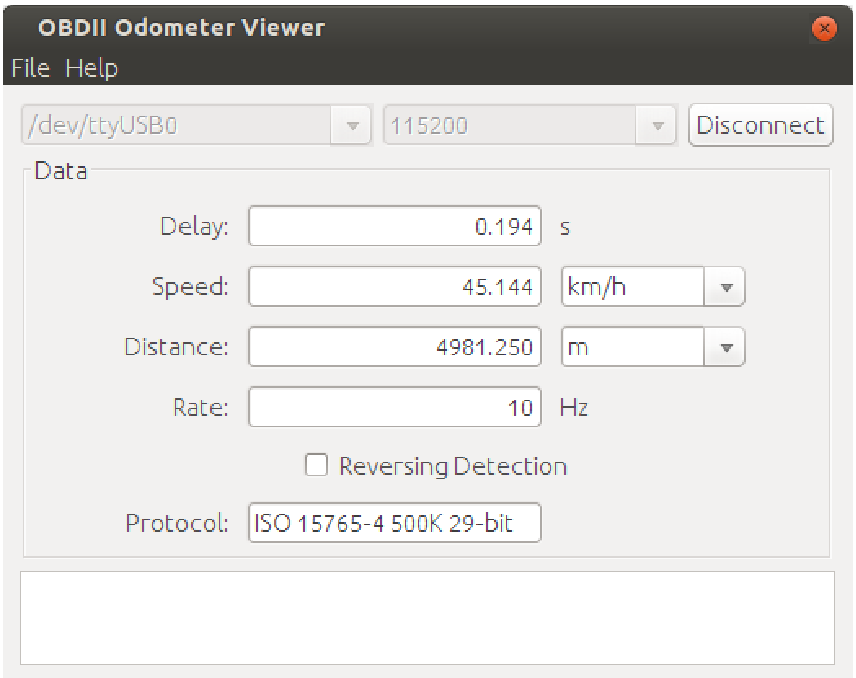

Illustration 7: OBDII Odometer Viewer screenshot

8.6 Firmware Updates

The OBDII Odometer features a bootloader to allow for firmware updates. The current and only version of firmware available for the OBDII Odometer is version 1.0. Once the next version is released this section of the reference manual will be updated with a changelog and instructions for updating firmware over the RS232 interface.

9 Advanced Navigation Packet Protocol

All communication to the OBDII Odometer is over the RS232 interface in the Advanced Navigation Packet Protocol (ANPP). The RS232 format is fixed at a baud rate of 115200, 1 start bit, 8 data bits, 1 stop bit, no parity and no flow control.

The Advanced Navigation Packet Protocol (ANPP) is a binary protocol designed with high error checking, high efficiency and safe design practices. It has a well defined specification and is very flexible. It is used across all existing and future Advanced Navigation products.

9.1 Data Types

The following data types are used in the packet protocol. All data types in the protocol are little endian byte ordering.

Abbreviation	Bytes	Also known as
u8	1	unsigned char, unsigned byte, uint8_t
s8	1	char, byte, int8_t
u16	2	unsigned short, uint16_t
s16	2	short, int16_t
u32	4	unsigned int, unsigned long, uint32_t
s32	4	int, long, int32_t
u64	8	unsigned long long, uint64_t
s64	8	long long, int64_t
fp32	4	float
fp64	8	double

Table 7: Data type abbreviations used in the ANPP

9.2 Packet Structure

The ANPP packet structure is shown in Table 8 and the header format is shown in Table 9. Example code can be downloaded from the software section.

Header				
Header LRC	Packet ID	Packet Length	CRC16	Packet Data

Table 8: ANPP Packet Structure

ANPP Header Format				
Field #	Bytes Offset	Data Type	Size	Description
1	0	u8	1	Header LRC, see section 9.2.1
2	1	u8	1	Packet ID, see section 9.2.2
3	2	u8	1	Packet Length, see section 9.2.3
4	3	u16	2	CRC16, see section 9.2.4

Table 9: ANPP header format

9.2.1 Header LRC

The header LRC (Longitudinal Redundancy Check) provides error checking on the packet header. It also allows the decoder to find the start of a packet by scanning for a valid LRC. The LRC can be found using the following:

LRC = ((packet_id + packet_length + crc[0] + crc[1])^0xFF) + 1

9.2.2 Packet ID

The packet ID is used to distinguish the contents of the packet. Packet IDs range from 0 to 255.

Within this range there are three different sub-ranges, these are system packets, state packets and configuration packets.

System packets have packet IDs in the range 0 to 19. These packets are implemented the same by every device using ANPP.

State packets are packets that contain data that changes with time, i.e. temperature. State packets can be set to output at a certain rate. State packets are packet IDs in the range 20 to 179.

Configuration packets are used for reading and writing device configuration. Configuration packets are packet IDs in the range 180 to 255. The OBDII Odometer does not have any configuration packets.

9.2.3 Packet Length

The packet length denotes the length of the packet data, i.e. from byte index 5 onwards inclusive. Packet length has a range of 0 – 255.

9.2.4 CRC

The CRC is a CRC16-CCITT. The starting value is 0xFFFF. The CRC covers only the packet data.

9.3 Packet Requests

Any of the state and configuration packets can be requested at any time using the request packet. See section 9.8.2.

9.4 Packet Acknowledgement

When configuration packets are sent to Spatial, it will reply with an acknowledgement packet that indicates whether the configuration change was successful or not. For details on the acknowledgement packet, see section 9.8.1.

9.5 Packet Rates

The packet rates are automatic and fixed on the OBDII Odometer. The Odometer Packet outputs at 10Hz. No other packets will automatically output. If the vehicle computer cannot keep up with a 10Hz rate, the speed may output at a slower rate. When the OBDII Odometer does not have a connection with a vehicle computer over OBDII no packets will output.

9.6 Packet Summary

Packet ID	Length	R/W	Name
System Packets			
0	4	R	Acknowledge Packet
1	-	W	Request Packet
2	1	R/W	Boot Mode Packet
3	24	R	Device Information Packet
5	4	W	Reset Packet
State Packets			
67	9	R	Odometer Packet
Configuration Packets			

Table 10: Packet summary

9.7 State Packets

The OBDII Odometer has only one state packet which is the Odometer Packet. The Odometer Packet outputs at 10Hz unless the vehicle computer is not able to keep up at this speed in which case it may output at a slower rate.

9.7.1 Odometer Packet

Odometer Packet				
Packet ID				67
Length				13
Field #	Bytes Offset	Data Type	Size	Description
1	0	fp32	4	Delay (s)
2	4	fp32	4	Speed (m/s)
3	8	fp32	4	Distance travelled (m)
4	12	u8	1	Flags, see section 9.7.1.1

Table 11: Odometer packet

9.7.1.1 Odometer Flags

Bit	**Description**
0	Reversing detection supported
1-3	Reserved (set to zero)
4-7	OBDII Protocol, see section 9.7.1.2

Table 12: Odometer flags

9.7.1.2 OBDII Protocol

Value	Description
0	OBDII protocol unknown
1	ISO 9141-2
2	ISO 14230-4 slow
3	ISO 14230-4 fast
4	ISO 15765-4 250K 11-bit
5	ISO 15765-4 250K 29-bit
6	ISO 15765-4 500K 11-bit
7	ISO 15765-4 500K 29-bit
8	SAE J1850 PWM
9	SAW J1850 VPW
10	SAE J1939
11	General Motors
12	Ford
13	European (Bosch)
14	Japanese
15	Reserved

Table 13: OBDII protocol

9.8 System Packets

9.8.1 Acknowledge Packet

Acknowledgement Packet				
Packet ID				0
Length				4
Field #	Bytes Offset	Data Type	Size	Description
1	0	u8	1	Packet ID being acknowledged
2	1	u16	2	CRC of packet being acknowledged
3	3	u8	1	Acknowledge Result, see section 9.8.1.1

Table 14: Acknowledge packet

9.8.1.1 Acknowledge Result

Value	Description
0	Acknowledge success
1	Acknowledge failure, CRC error
2	Acknowledge failure, packet size incorrect
3	Acknowledge failure, values outside of valid ranges
4	Acknowledge failure, system flash memory failure
5	Acknowledge failure, system not ready
6	Acknowledge failure, unknown packet

Table 15: Acknowledge result

9.8.2 Request Packet

Request Packet				
Packet ID				1
Length				1 x number of packets requested
Field #	Bytes Offset	Data Type	Size	Description
1	0	u8	1	Packet ID requested
+				Field 1 repeats for additional packet requests

Table 16: Request packet

9.8.3 Boot Mode Packet

Boot Mode Packet				
Packet ID				2
Length				1
Field #	Bytes Offset	Data Type	Size	Description
1	0	u8	1	Boot mode, see section 9.8.3.1

Table 17: Boot mode packet

9.8.3.1 Boot Mode Types

Value	Description
0	Bootloader
1	Main Program

Table 18: Boot mode types

9.8.4 Device Information Packet

Device Information Packet				
Packet ID				3
Length				24
Field #	Bytes Offset	Data Type	Size	Description
1	0	u32	4	Software version
2	4	u32	4	Device ID
3	8	u32	4	Hardware revision
4	12	u32	4	Serial number part 1
5	16	u32	4	Serial number part 2
6	20	u32	4	Serial number part 3

Table 19: Device information packet

9.8.5 Reset Packet

Reset Packet				
Packet ID				5
Length				4
Field #	Bytes Offset	Data Type	Size	Description
1	0	u32	4	Verification Sequence (set to 0x21057A7E)

Table 20: Reset packet