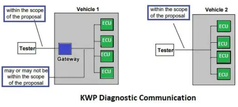
KWP-2000 Protocol

The KWP-2000 Protocol, commonly named as KWP2000, is a diagnostic Communication protocol for On-Board vehicle Diagnostics applied for OEM. There are two primary International Standardized documents such as ISO 9141, ISO 14230 covered for performing the OBD Communications on K-line between Tester and vehicle.ISO 14230 specifies common requirements of diagnostic services which allow a tester to control diagnostic functions in an on-vehicle Electronic Control Unit.

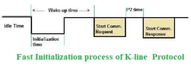


KWP Diagnostic Communication

KWP Diagnostic Communication

The ISO 9141 is a bidirectional serial communication detail which specifies K-line is used for communication and initialization; the L-line (optional) is used for initialization only. The Special cases are node-to-node-connection, this means there is only one ECU on the line which also can be a bus converter. ISO 14230-2 specifies the Data link layer and ISO 14230-3 includes all the definitions which are necessary to implement the Diagnostics services as per KWP-2000 Protocol.

The Keyword Protocol physical layer may be used as a multi-user bus system, it is necessary to maintain or manage arbitration and bus management system. In below figure described the Vehicle Diagnostics architecture of KWP-2000 Protocol which specifies communication among Tester and ECU network with Gateway or Without Gateway access.



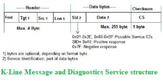
KWP Communication Timing Diagram

In KWP-2000 Protocol there is no continuous communication must be initialized without being first requested by the tester which means that Wake-Up at 5 baud per second. The 5 Baud address byte is transferred from the Tester on K-line and on L-line. After sending the 5 baud address byte the tester will maintain L-Line on the High level. After the Wake-up procedure, all ECU's are initialized shall use a baud rate of 10,400 Baud for initialization and Communication.

At first, the tester transmits a Wakeup pattern on K-line and L-line synchronously in KWP-2000 Protocol. The pattern has begins after an idle time on K-line with a low time of initialization value. The Tester transmits the first bit of the start communication service after a time of wake-up time following the first falling edge, as shown in the below figure.

KWP Communication

KWP-2000 Protocol Message Format



KWP Protocol Message Format

Search Your Idea Here...

Piest Forum - Android App Link

Gift A Cup Of Coffee To PiEmbSystech

JOIN TELEGRAM FOR TECH. DISCUSSION

Recent Posts

- Understanding Inheritance in CPP Programming Language
- Objects of a Class in CPP Language
- Classes in CPP Programming Language
- How Crystal Oscillators Work in Microcontrollers: Ensuring Precise Timing and Stable Performance
- Download and Install Turbo C++ Compiler

Archives

Select Month

Embedded Research Forum

Table Of Contents

8051 Microcontroller
8085 Microprocessor
8086 Microprocessor
About Us
Account
Adaptive AUTOSAR
Advanced driver assistance systems (ADAS)
Arduino
ARINC Protocol
ARM Microcontroller
Artificial Intelligence
Assembly Language
Automotive Architecture
Automotive BAP Protocol
Automotive ECU
Automotive Protocols
Automotive Safety
AUTOSAR
AUTOSAR DCM
AVR Microcontroller
Basic Electronics
Basic Understanding Of VLSI
BlueTooth Protocol
Boot Loader
ByteFlight Protocol
C Plus Plus (CPP) Tutorial
C-Language
CAN Protocol
CAN-FD Protocol
CAN-TP Protocol
CanaLyzer
Canoe
CAPL Language
ChibiOS/RT
CMSIS-RTOS
Contact Us
Contiki RTOS
Cookie Policy
CPU Design
Dashboard
Disclaimer
DMC
DoCAN Protocol
DoIP Protocol
DSRC Protocol
eCos RTOS
Edit
Embedded Linux
EtherNet Protocol
FlexRay Protocol
FlexRay Transport Protocol (ISO 10681-2)
Free RTOS
Guest Post
Home
HTTP (Hypertext Transfer Protocol): An Overview of the Internet's Most Widely Used Protocol
Hw/Sw Interface
I2C Protocol
INTEGRITY Operating System
IoT
ISO-15031 Protocol
K-Line Protocol
KWP-2000 Protocol
LIN Protocol
Linux Basics
Linux Device Driver
Linux IPC
Linux Kernel
Linux System Architecture
Login
Long Range Wide Area Network (LoRaWAN) Protocol
Mastering MIPI I3C Protocol: A Comprehensive Guide to Efficient Communication Between Devices
Mbed OS: Arm's Open-Source OS for IoT Devices
Message Queuing Telemetry Transport (MQTT) Protocol
Microcontroller
MODBUS Protocol
MOST Protocol
Motor Design
Nucleus RTOS
OBD-II
Operating System
Order Received
OSAL
OSEK
Payment
Power Electronics
PowerPC Processor
Privacy Policy
QNX RTOS
Raspberry-Pi
Register
Resistor
RIOT Operating System
Robotics
RT-Thread RTOS
RTLinux RTOS
RTOS Concept
SAE J1708 Protocol
SAE-J1939 Protocol
SENT Protocol
SPI Communication Protocol: A Comprehensive Guide to Serial Peripheral Interface
Subscription
Terms and Conditions
Thank You
ThreadX RTOS: A Lightweight and Scalable RTOS for Embedded Devices
TinyOS
Transmission Control Protocol (TCP/IP)
UART Protocol
uC/OS RTOS
UDS Protocol
Understanding of Internet Protocol (IP) - The Backbone of the Internet
USB Protocol
User Datagram Protocol (UDP)
V2X Communication
VxWorks RTOS: A High-Performance Real-Time Operating System for Embedded Systems
Wi-Fi Protocol
Windows OS: An Evolution in GUI Based OS
XBEE Protocol
XCP Protocol
Zephyr RTOS

Automotive Electronics
Computer Science
Electronics Technology
Linux System
Programming Language
- C
- C++
- Python

Robotics Technology
VLSI