SMART OBDII INTERFACE ADAPTER

[1] Peter Dzhelekarski, [2] Dimiter Alexiev

Faculty of Electronic Engineering and Technologies, Technical University of Sofia,
8 Kliment Ohridski Str., 1000 Sofia, Bulgaria, phone: [2] +359 2 965 3677,
e-mail: [1] pid@tu-sofia.bg, e-mail: [2] dalex@tu-sofia.bg

The vehicle on-board diagnostics system (OBDII) provides a serial diagnostic link for communication with scan tools. There are 4 diagnostic interfaces: ISO 9141-2, KWP 2000, SAE J1850 and ISO 15765. A PC-based scan tool consists of an interface adapter and specialized diagnostic software. This paper presents the implementation of a smart OBDII interface adapter. The adapter is build of a protocol converter, interface transceivers and protective devices. The full schematic circuit of the adapter is presented and explained in detail. The protocol converter (MCU) processes and transfers diagnostic messages. It communicates with PC via serial port using ASCII messages. The software for the protocol converter has been developed in C language. A state diagram of software is presented. The functions performed by the software for different interfaces are explained in detail. The adapter is successfully verified in practice.

Keywords: OBD, scan tool, interface adapter, K-Line, CAN

1. INTRODUCTION

The OBDII (On-Board Diagnostics II) system represents the capability of vehicle electronic systems to diagnose itself and its components. The output from the OBD system is presented to the driver as a warning light with an engine symbol. When a fault has been detected, a diagnostic trouble code is set and stored in the computer memory. OBD provides a serial diagnostic link for communication with external test equipment (scan tool). One of the following communication protocols is used [2, 5]:

1) ISO 9141-2 – baud rate 10.4 kbit/s;
2) ISO 14230 (KWP 2000 – Keyword protocol 2000) – the same baud rate;
3) SAE J1850 (Class B data communication interface) – rates 10.4/41.6 kbit/s;
4) ISO 15765 (CAN – Controller area network) – baud rates 250/500 kbit/s.

The ISO 9141-2 and KWP 2000 interfaces are the most common diagnostic interfaces in Europe (KWP 2000 is newer). They share the same physical interface, called K-Line interface. SAE J1850 is mostly used in the American vehicles. It has two alternative physical layers: VPW and PWM. CAN is mainly used in new cars with model year after 2003.

The scan tool should obtain and display diagnostic information stored [5]. A PC-based scan tool consists of an interface adapter and specialized diagnostic software. The interface adapter provides a connection between the vehicle and the PC. The connection to PC is most often via RS 232 interface. One ore more diagnostic interfaces can be supported. In case only K-Line interface is used, the adapter can be implemented as a simple electrical converter to RS 232. When CAN or J1850 interfaces are used, the adapter should contain a protocol converter, implemented by means of MCU.

The protocol converter processes and transfers diagnostic messages according to the data link layer of the respective protocol. If the protocol converter is utilized for processing of K-Line messages, this leads to improved message timing and decreased PC load as well [1, 4].

This paper presents a method for implementation of a smart OBD interface adapter for connection between all 4 diagnostic interfaces and the PC serial port. The adapter is called *smart* because of the presence of MCU. The adapter is build of a protocol converter (MCU), interface transceivers and protective devices (Fig. 1). This method of implementation uses an intermediate conversion of the electrical levels to TTL level that is necessary for connection of MCU. The protective devices suppress the overvoltages, coming from the vehicle lines. The interface transceivers are two-way converters of electrical levels between TTL and the respective interface signals.

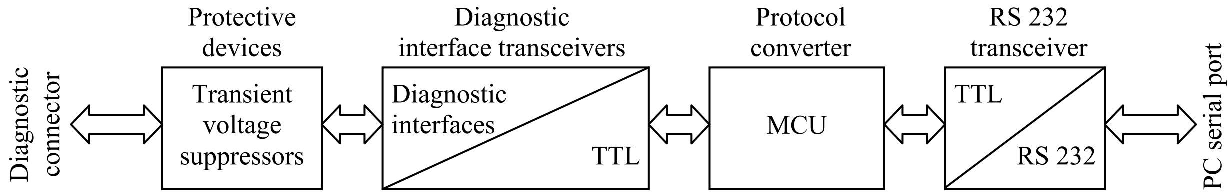


Fig. 1. General block diagram of OBDII interface adapter

2. Schematic Circuit of Interface Adapter

The schematic circuit of the adapter shall conform to the requirements for the physical layer of diagnostic interfaces and RS 232.

The full schematic circuit of the interface adapter is presented at Fig. 2. The signals coming from the vehicle diagnostic connector are fed to connector *J1* via a standard diagnostic cable. The protective devices of the circuit are *D1* (TVS) and *D3-D10* (voltage delimiters). The adapter is powered by the vehicle battery. A power supply +5 V (*U1*) necessary for MCU, RS 232- and CAN transceivers is provided.

The K-Line transceiver is implemented by an open collector comparator (*U3A*) and transistor inverter stages (*Q1*/*Q2*). The *RxK* signal is for receiving information from *K*-line. The $\overline{TxK}/\overline{TxL}$ signals are for transmitting information to *K*/*L* lines. The comparator has a reference voltage of +5V and is used for reading information from *K* line. The transistors are used for sending information on *K*/*L* lines.

At this stage of project the J1850 transceiver is not implemented. It is a subject of future work. Nevertheless five J1850 communication signals are provided: for VPW physical layer – *TxVPW*, *RxVPW*; for PWM physical layer – *RxPWM*, $\overline{TxPWM(+)}/\overline{TxPWM(-)}$.

The CAN transceiver *U2* connects with 2 signals to the protocol converter *U5*: *TxCAN* and *RxCAN*. *U2* has an AC termination circuit (*R2-C4/R3-C5*) according to ISO 15765-4 requirements. *U2* is enabled by the signal *EN_CAN* [3].

The MCU *U5* is a 28-pin device with the following features: PIC18 8-bit instruction set (optimized for C compilers), 16 kB Flash memory, 768 B RAM, 256 B EEPROM, ECAN module, EUSART module, etc [6]. The selection of MCU is based on the number of pins, memory size and communication capabilities. The clock speed F_{OSC} = 20 MHz conforms to the required CAN module configuration (see 3.2).

One driver and one receiver are utilized from the RS 232 transceiver *U4*. They are connected between *RC6/TX* and *RC7/RX* pins of MCU and *TxD/RxD* signals of *J2*.

3 status LEDs are provided: for power-on state (*D2*) and for transmission/ reception of OBD data messages (*D11*/*D12*).

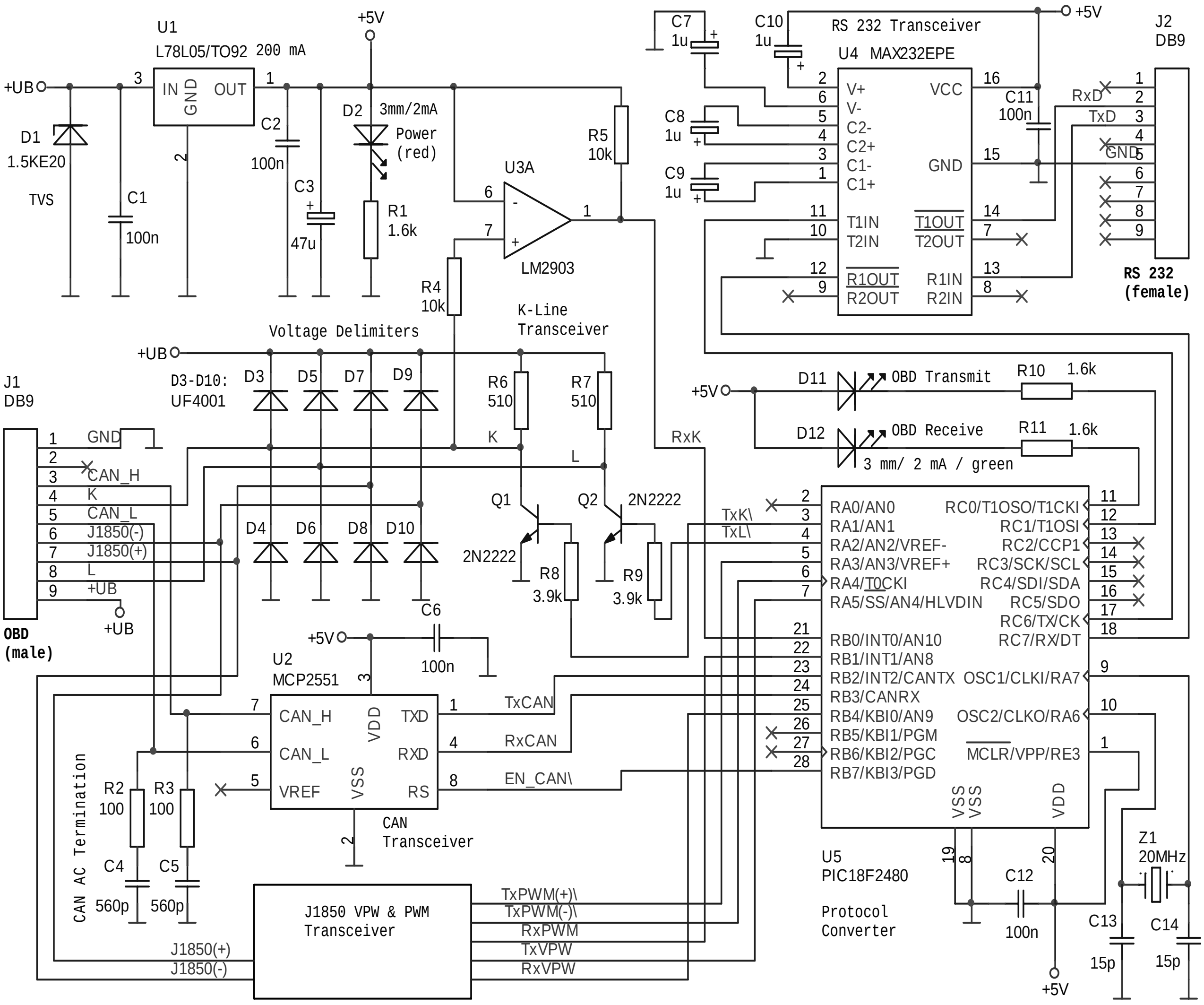


Fig. 2. Schematic circuit of OBDII interface adapter

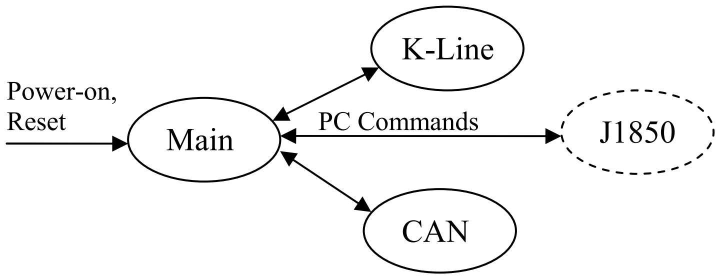
3. PROTOCOL CONVERTER SOFTWARE

The software for the protocol converter has been developed in C programming language using Microchip MPLAB® C18 C Compiler.

The operation of the protocol converter is controlled by PC via USART. The data is exchanged in form of ASCII encoded messages using baud rate of 38.4 kBaud. The first byte of a message represents the type of the command. The rest of bytes are hex bytes, each transmitted as a pair of ASCII bytes. The messages are terminated with

'\n' ($0A) characters. This way of communication allows longer messages with sizes limited only by the available RAM of MCU. This is necessary for communication using ISO 14230-3 protocol where the diagnostic message length is up to 260 bytes.

A state diagram of the software is shown in Fig. 3. Each state is implemented as a separate program loop. In each loop the program checks if a valid USART message has been received. On successful reception the requested operation is performed and a confirmation message is sent back to PC. A message containing an error code is sent back if an error has occurred or the command can not be performed. The program states are changed by commands from the PC. When the program is in main state, information about the version and the serial number of the protocol converter can be queried. The K-Line and CAN states will be described in detail. J1850 state is not implemented at present time.



State	Protocol
K-Line	ISO 9141-2 / KWP 2000
J1850	SAE J1850
CAN	ISO 15765

Fig. 3. State diagram of protocol converter software

3.1 K-Line State

When the program is in K-Line state, the protocol converter provides and performs the following functions:

- 5-baud initialization with specified address byte;
- Fast initialization with specified request message;
- Message transfer of specified request message and response message(s);
- Periodical sending of specified keep connection message.

All these functions perform sending and receiving bytes on *K* line. For this reason a simple software USART for pins $RxK/\overline{TxK}$ has been implemented.

The sending of bytes utilizes *Timer2* with period register *PR2* for generating the bit time. The $\overline{TxK}$ signal is driven according to the bit values of the respective byte.

The reception of bytes begins with the interrupt *INT0* on a falling edge of *RxK* (start bit). A dedicated interrupt service routine samples the *RxK* signal three times and takes the prevailing bit value using the same bit timer. By default the bit time is 96 μs (corresponding to standard baud rate of 10.4 kBaud) but it can be changed.

The 5-baud initialization [2] begins with sending an address byte using a timer for milliseconds (*Timer1* with *CCP1*) for generating the bit time (200 ms). The $\overline{TxK}\,/\,\overline{TxL}$ signals are driven simultaneously according to the bit values of specified address byte. Besides working with a fixed preset baud rate, this function can detect automatically the baud rate from the synchronization pattern ($55) and modify the period register accordingly after that.

The fast initialization [2] begins with a *wake-up pattern* (25 ms low level; 25 ms high level) sent simultaneously to $\overline{TxK} / \overline{TxL}$ signals. After that a request message is sent and responses are received using the next function (message transfer).

The message transfer [2] is performed according to the message timing requirements which are specified in the data link layer protocols. The transfer sequence is the following: at first the protocol converter sends to the vehicle a request message received from PC; then response bytes are received and grouped into one or more response messages. The reception is implemented by two nested loops. The outer loop is for message responses and the inner loop is for message bytes.

The periodical sending of specified message prevents the connection from automatic disconnection by ECU if there is no activity on the bus for more than 5 s. The message need to be set only once by PC. If there is an active connection the message is sent every 4 seconds after the last activity on *K* line.

The smart interface adapter operating in K-Line state has the following advantages compared to the simple K-Line interface adapter:

- Precise message timing;
- Precise wake-up pattern generation;
- Automatic baud rate detection;
- Automatic keeping of connection;
- Decreased PC load.

3.2 CAN State

When CAN state is entered at first the internal ECAN module is configured according to ISO 15765. This protocol defines requirements to vehicle diagnostic systems implemented on CAN communication link. A network layer is specified which allows the transfer of messages up to 4095 bytes long by means of data segmentation. The length of data bytes is fixed to 8 bytes. The first byte is always the protocol control information (PCI). Data segmentation is used only when the length of message to be transmitted exceeds 7 bytes.

There are 4 allowed configurations depending on baud rate and ID (identifier):

- Baud rate 250 kbit/s with standard ID (11-bit);
- Baud rate 250 kbit/s with extended ID (29-bit);
- Baud rate 500 kbit/s with standard ID (11-bit);
- Baud rate 500 kbit/s with extended ID (29-bit).

Each particular configuration can be activated by a command from PC. The bit timing configuration depending on the baud rate is given in Table 1. Depending on the type of ID, two acceptance filtering configurations are used (see Table 2). The dedicated filter/mask registers of the ECAN module are configured so that only messages with the required IDs are received. According to ISO 15765-2 the standard IDs use *normal addressing* and the extended IDs use *normal fixed addressing*.

Table 1. Bit timing configuration of ECAN module

Baud rate	**F_{OSC}**	**F_Q / T_Q**	**Timing Segment 1**	**Timing Segment 2**	**Synchronization**

					Jump Width
250 kbit/s	20 MHz	5 MHz / 200 ns	15 x T_Q	4 x T_Q	3 x T_Q
500 kbi/s	20 MHz	10 MHz /100 ns	15 x T_Q	4 x T_Q	3 x T_Q

Table 2. Acceptance filtering configuration of ECAN module

ID Type	**ID Byte 3** (bits 28 through 24)	**ID Byte 2** (bits 23 through 16)	**ID Byte 1** (bits 15 through 8)	**ID Byte 0** (bits 7 through 0)
11-bit	–	–	0000 0111b	1110 1XXXb[1)]
29-bit	18h	DAh	F1h	XXh[2)]

Legend: 'X' means do not care.
[1)] XXXb is the number of ECU: from 0 to 7.
[2)] XXh is the address of ECU: from 00h to FFh.

The CAN state provides a function for sending a specified request message and receiving response frames which are part of segmented or single-frame messages. The handling of network layer is not a task of protocol converter. The converter only transfers accepted frames via USART to PC in the order in which they are received. However the protocol converter observes the timeouts specified in the network layer.

4. RESULTS

The presented smart OBDII interface adapter is verified successfully in practice for K-Line and CAN interfaces. All functions provided in K-Line state, including automatic baud rate detection, are tested for ISO 9141-2 and KWP 2000 protocols. The precise and accurate message timing is proved by reliable reception of multiple message responses. All 4 possible configurations in CAN state are tested by transfer of diagnostic messages.

5. CONCLUSION

A method for implementation of smart OBDII interface adapter for connection between vehicle diagnostic interfaces and a PC serial port is presented. The interface adapter conforms to the physical layer and the data link layer protocols. It is built of a protocol converter (MCU), interface transceivers and protective devices. The protocol converter processes diagnostic messages according to the respective data link layer protocol. It communicates with the PC via serial port using ASCII messages. For K-Line interface the adapter provides functions for initialization with automatic baud rate detection, message transfer and periodical sending of keep connection messages. For CAN interface the adapter can be configured in all 4 required modes. A function for message transfer is provided. The adapter is successfully verified in practice.

6. REFERENCES

[1] *ELM 327 OBD to RS232 Interpreter*. Elm Electronics, ELM327DSA, 2005.
[2] Dzhelekarski, P. and D. Alexiev. *Initializing Communication to Vehicle OBDII System*. Proceedings of the Fourteenth Int. Conference ELECTRONICS'05, Book 3, pp. 46-52, 2005.
[3] *MCP2551 High-speed CAN Transceiver*. Microchip, DS39637A, 2004.
[4] *OE90C2600 EOBD/OBDII to RS232 Gateway*. Özen Elektronik, 2004.
[5] Özen, B. *Auto-Diagnose mit OBD-2*. Elektor, 7-8/2005, pp. 14-18.
[6] *PIC18F2480/2580/4480/4580 Enhanced Flash Microcontrollers*. Microchip, DS39637A, 2004.