Chapter 1 Functionality and Element Position Map of Demoboard

This demoboard integrates common vehicle bus and functional test, allowing user to fully understand the handling methods of all vehicle bus messages and all kinds of physical bus connections.

The diagram of demoboard is included in Document .

The diagram file: <**ET7190_Demoboard Diagram.xps**.>. under this catalogue.

1.1 Development Tool Support

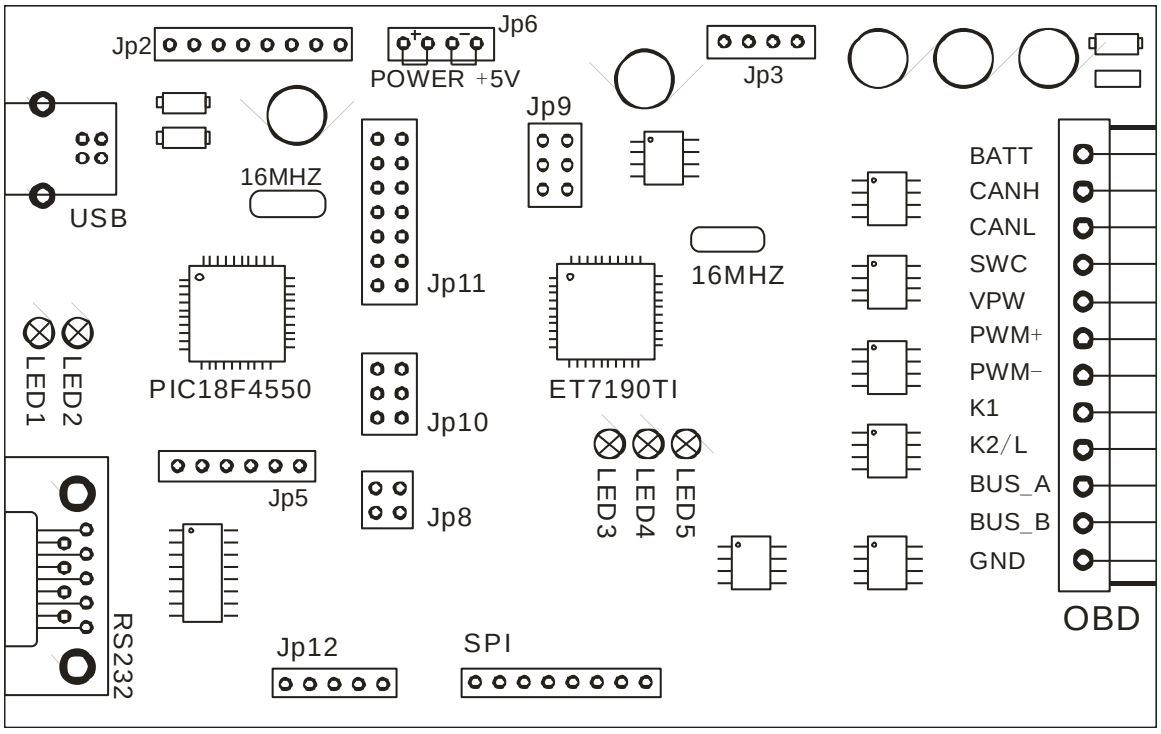
- J1850VPW, J1850PWM complete functions and bus data monitoring.
- CAN ISO15765/J1939/TP2.0 complete functions and bus data monitoring.
- IS014230 ISO9141 KW1281 J1708 LIN complete functions and bus data monitoring.
- Well-functional K-line logic: When diagnostic tester communicates with ECU (Electronic Control Unit), the initialization timing of specific diagnostic tester can be easily obtained. Several actual initialization processes are contained under UserFile Catalogue, which is ready to view by opening files.
- Full-functional OBD for the diagnosis of ECU simulation function. Any application protocol of ISO9141-2/ISO14230(KWP2000) /J1850VPW/J1850PWM /ISO15765 can be simulated. The initialization process and respond data are fully user-defined, every module on the vehicle can be simulated, when ECU is simulated on ET7190 demoboard, work timing is accurate and normative.
- A computer can simultaneously connect to several demoboards, open several applications, and independent equipment operation can be selected via each of these applications.

1.2 Demoboard-supported Vehicle Bus

- Double-line high-speed CAN, up to 1MBPS.
- Support ISO15765/SAE J1939 /TP2.0 or any protocol based on double-line CAN.
- Single-line CAN
- Support SAE J2411 / GMW3089 / GMW3110 or related
- J1850 VPW Bus
- J1850 PWM Bus
- K1/K2 Bus, compatible to SCI Bus, support ISO14230(KWP2000)/ISO9141-2/KW1281/LIN1.x/LIN2.x/SCI (SAE J2610)/SAE J2534 or related
- SAE J1708 Bus: Support SAE J1708/J1587 serial communication protocol for commercial vehicle

Note: Demoboard supports the vehicle bus in 12V/24V battery voltage

1.3 Element Position Map of Demoboard



Description of function of ports:

JP2: PD7..0 IO port of PIC18F4550 MCU.

JP3: Output port of ET7190 P0/P1/P2/P3 for user's extended test.

JP5: ICSP debug programming port of PIC18F4550 MCU.

JP6: 5V power input (When USB port is not occupied, the demoboard can be powered from here)

JR8: TX/RX wire jumper that connects RS232 and ET7190.

JP9: ET7190 UI mode selection. (For the selection of Baud rate)

JP10: Connected to ET7190 via UART port of PIC18F4550 microprocessor.

JP11: Connected to ET7190 via SPI port of PIC18F4550 microprocessor.

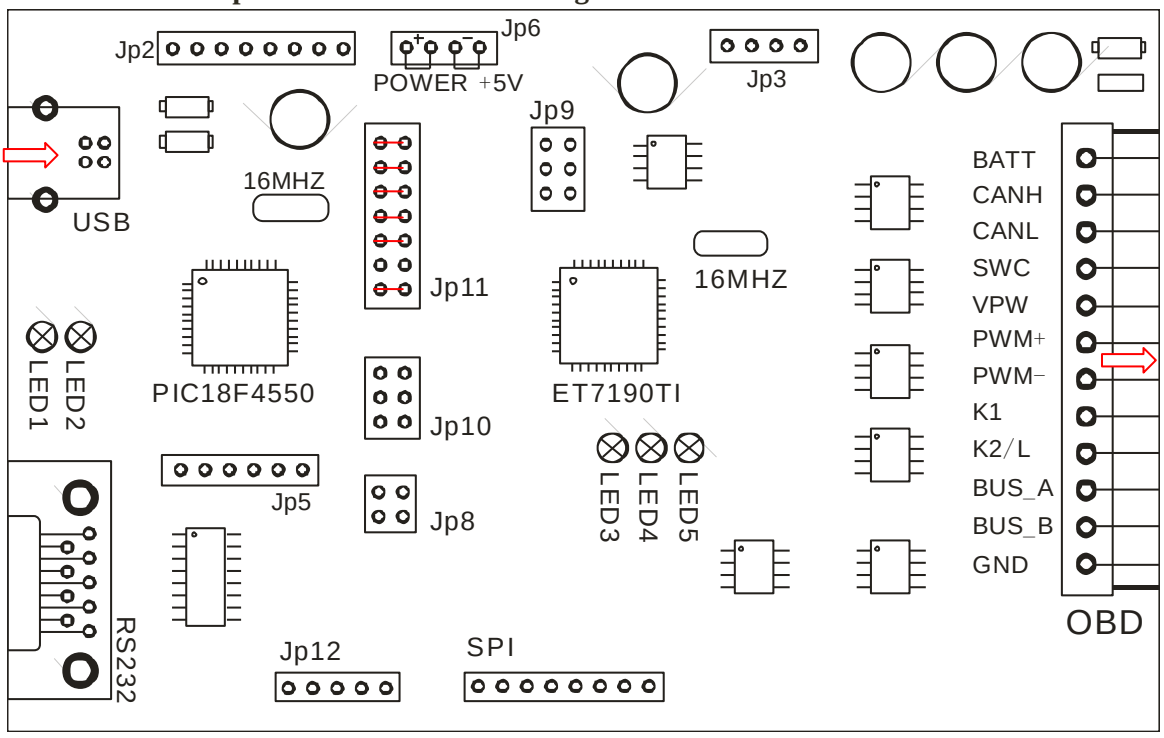
JP12: ET7190 TX RX pin, from which user board or Bluetooth module can connect to ET7190 UART mode via this port for development.

SPI: ET7190 SPI pin, from which user board can connect to ET7190 SPI mode for development.

RS232: Directly connected to PC serial port or USB virtual serial port. **OBD**: Vehicle bus connector.

1.4 Wire Jumper Setting in the Applications of Demoboard

1.4.1 Wire Jumper and Power Source Using ET7190Kits tool

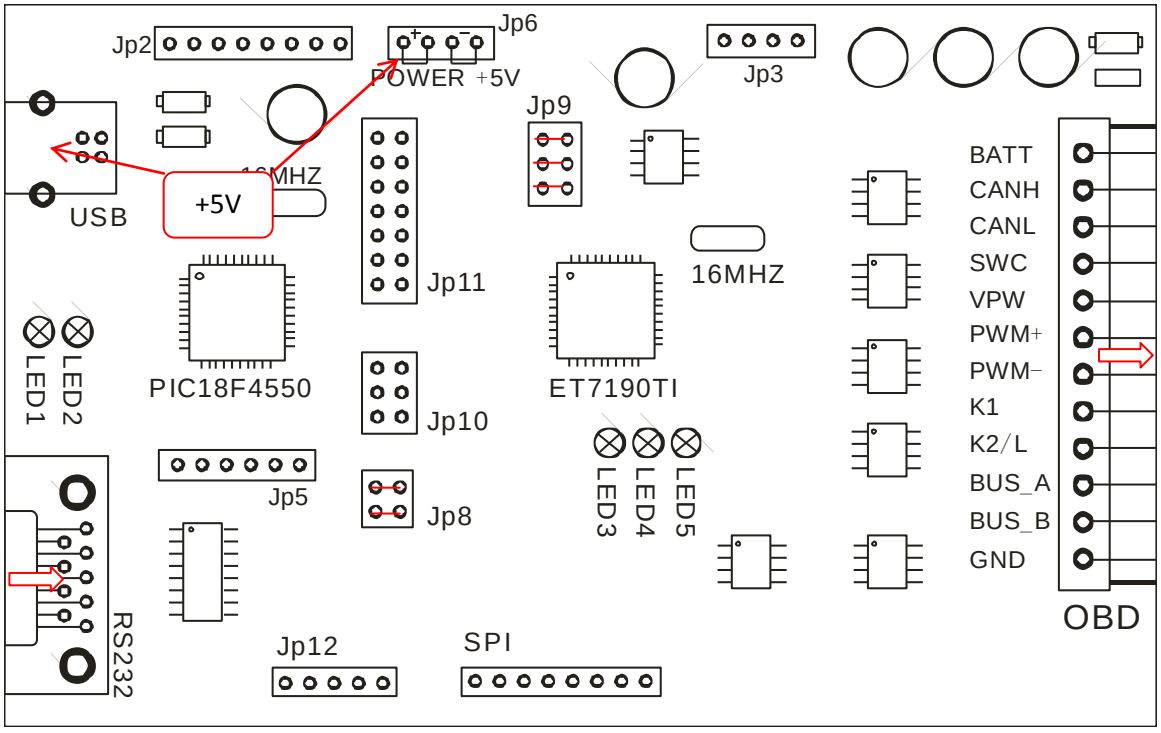


Description of Connections:

- ET7190Kits tool connects to ET7190 chip via USB—>PIC18F4550 SPI interface. Wire jumper is shown as above, when ET71930 MOD pin is in high level, SPI user interface connects to PIC18F4550, /INT pin connects to PIC18F4550 external interrupt pin.
- In such a case, Demoboard is powered by USB port and JP6 doesn't need any external power source.
- The bus should connect to BATT power source (12Vor 24V) when using K1/K2/VPW/SWC. If K1/K2/VPW/SWC bus is not used, but CAN/J1850PWM/J1708 (BUS_A/BUS_B) bus is only used, BATT power source may not be connected.
- LED1 is 5V power indicator. LED2 blinks when it reads and writes number at USB port.
- LED3 is BATT power indicator. LED4/LED5 blinks in writing or reading ET7190 data buffer respectively, indicating the current status of bus.
- The port used by OBD can directly connect to vehicle bus. (No element is needed when connecting it to vehicle bus)

Note: If any of SWC/VPW/K1/K2 bus is used, then Demoboard GND must connect to vehicle ground wire. If CANH/CANL J1850PWM(+ -) J1708(A/B) bus is only used, then GND may not connect, because the latter three buses are differential buses, which means that communication is ready with no common ground.

1.4.2 RS232COMM Example Setting and Wire Jumper



Description of connections:

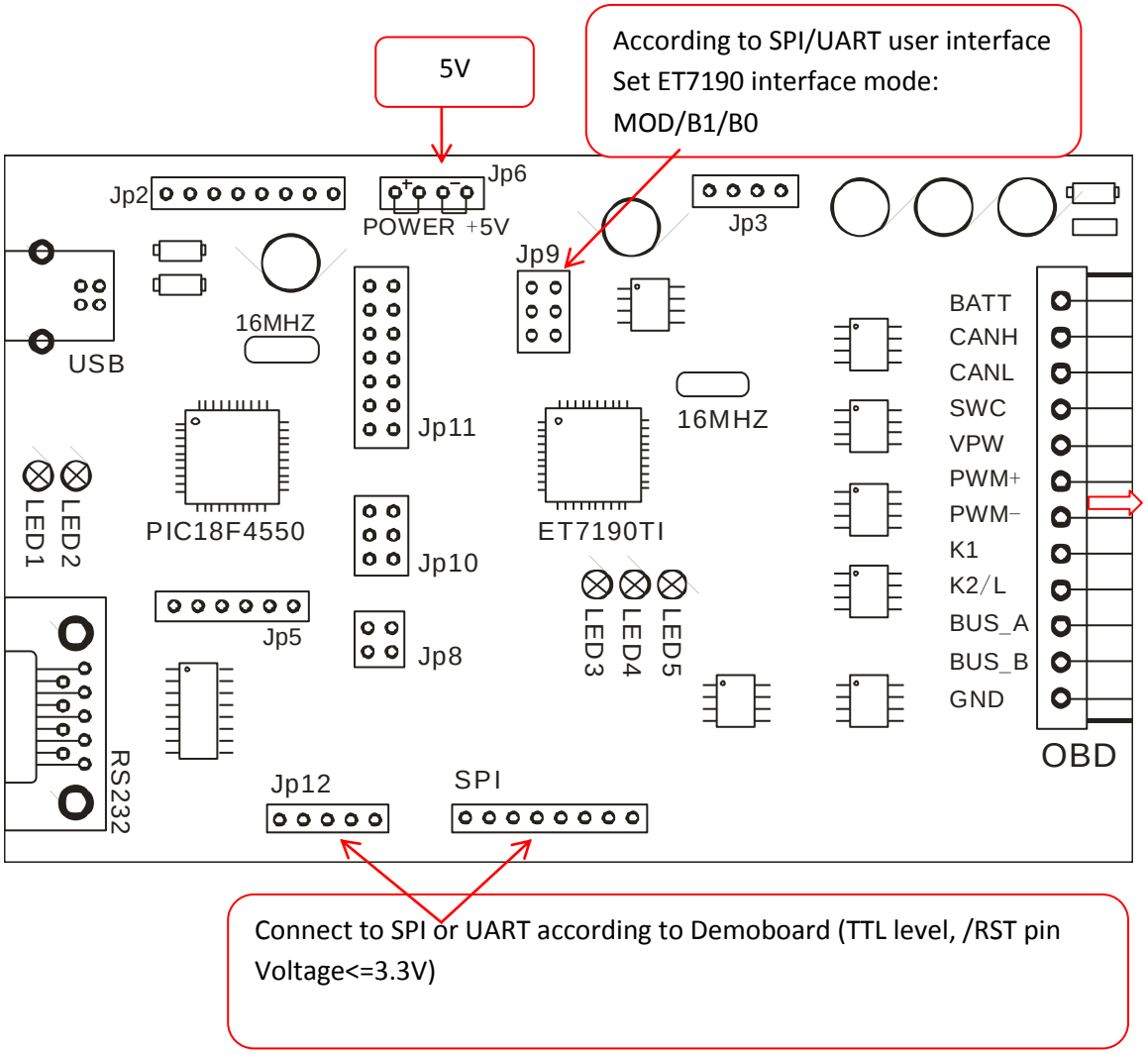
- In using RS232COMM example, ET7190 chip is connected via PC serial port. Wire jumper is as shown above, when ET71930 MOD pin is in low level, B1:B0=00, ET7190 reset Baudrate in JP9 is 9,600bps, which is increased to higher communication Baudrate by changing DRVUBRG in the program.
- Demoboard is powered by either USB port or JP6 with no need to connect to both power sources.
  JP12 5V/3.3V port is output, which can power other modules
- In this case PIC18F4550 doesn't work, if USB is used, it only functions as the power source.
- In using K1/K2/VPW/SWC, bus must connect to BATT power source (12Vor 24V). If K1/K2/VPW/SWC bus is not used, but CAN/J1850PWM/J1708(BUS_A/BUS_B) bus is only used, BATT power source may not be connected.
- LED3 is BATT power indicator. LED4/LED5 blinks in writing or reading ET7190 data buffer respectively, indicating the current status of bus.
- The port used by OBD can directly connect to vehicle bus. (No element is needed when connecting it to vehicle bus)

Note: If any of SWC/VPW/K1/K2 bus is used, then Demoboard GND must connect to vehicle ground wire and it must connect to BATT. If CANH/CANL J1850PWM(+ -) J1708(A/B) bus is only used, then GND and BATT may not connect, because the latter three buses are differential buses and 5V power source is only used, which means that communication is ready with no common ground.

Note: Wire jumper must not be random, or it may causes input and output short circuit to damage IC chip

1.4.3 Setting of User Target Board Connected to Demoboard

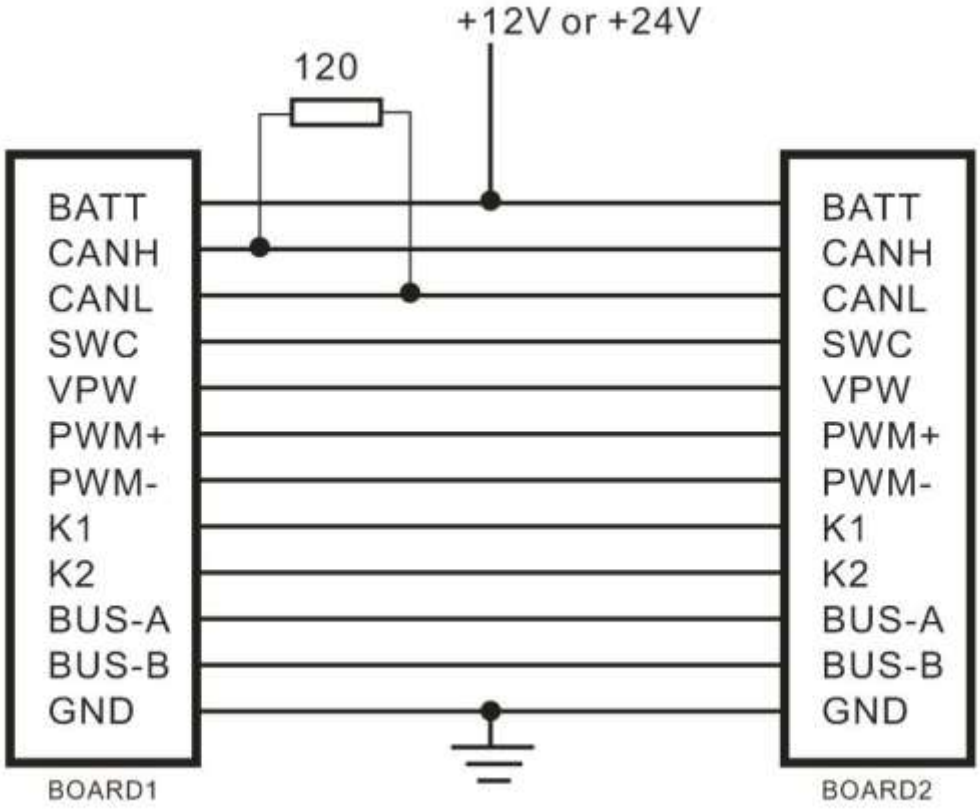
All wire jumpers of JP8/JP10/JP11 are disconnected, ET7190 UI signal wire is connected to Demoboard via SPI or JP12(UART) for the secondary development.



- JP12 may connect to Tx/RX pin at UART port of user MCU, or connect to the Wireless Bluetooth serial port. ET7190 TX pin outputs 3.3V as the high level. TX/RX pin may directly connect to 3.3V or 5V system.
- The maximum allowed voltage of RST pin at SPI port is 3.3V, other pins may directly connect to 3.3V or 5V system.
- Wiring of OBD interface connected to vehicle bus is the same.
- In this case RS232 and USB interface don't work.
- Three wire jumpers of JP9 are disconnected when using SPI to connect to user board, where MOD=1 and chip enables SPI function.

In using JP12 to connect to user board, JP9 MOD must be short-connected, MOD pin input is 0, B1:B0 selects the initial reset Baud rate. (Pull-up resistor is mounted in ET7190 chip of MOD, B1, B0 pin)

1.5 Interconnection between Two Demoboards



- With the interconnection between two Demoboards, intercommunication is ready when developing the application for the convenience of program debugging.
- One board may simulate all ECU functions using ET7190Kits tool, the other may process the development of diagnostic program.
- If demoboard bus simultaneously connects to vehicle bus, 120 Resistor as given above must not be connected. (Two CAN terminal resistors have been mounted on the vehicle), if only two Demoboards are interconnected, resistor must be connected. Only a resistor is needed for the test to ensure CAN bus with loop for normal communication.

Chapter 2 Use the Development Tool ET7190Kits

This software is used to monitor bus data and simulate ECU work, which also serves as the necessary tool for the ex-factory diagnostic protocol in reverse study. ET7190Kits development tool has the main functions as follows:

- J1850VPW, J1850PWM complete functions and bus data monitoring.
- CAN ISO15765/J1939/TP2.0 complete functions and bus data monitoring.
- IS014230 ISO9141 KW1281 J1708 LIN complete functions and bus data monitoring.
- Well-functional K-line logic: When diagnostic tester communicates with ECU (Electronic Control Unit), the initialization timing of specific diagnostic tester can be easily obtained. Several actual initialization processes are contained under UserFile catalogue, which is ready to view by opening files.
- Full-functional OBD for the diagnosis of ECU simulation function. Any application protocol of ISO9141-2/ISO14230(KWP2000) /J1850VPW/J1850PWM /ISO15765 can be simulated. The initialization process and respond data are fully user-defined, every module on the vehicle can be simulated, when ECU is simulated on ET7190 demoboard, work timing is accurate and normative.
- A computer can simultaneously connect to several demoboards, open several applications, and independent equipment operation can be selected via each of these applications.

2.1 Wire Jumper Setting Using ET7190KITS Software

Wire jumper and power source are connected as per Chapter 1 Section 1.3.1.

2.2 ET7190KITS Files and Installation Method

2.2.1 ET7190KITS Files

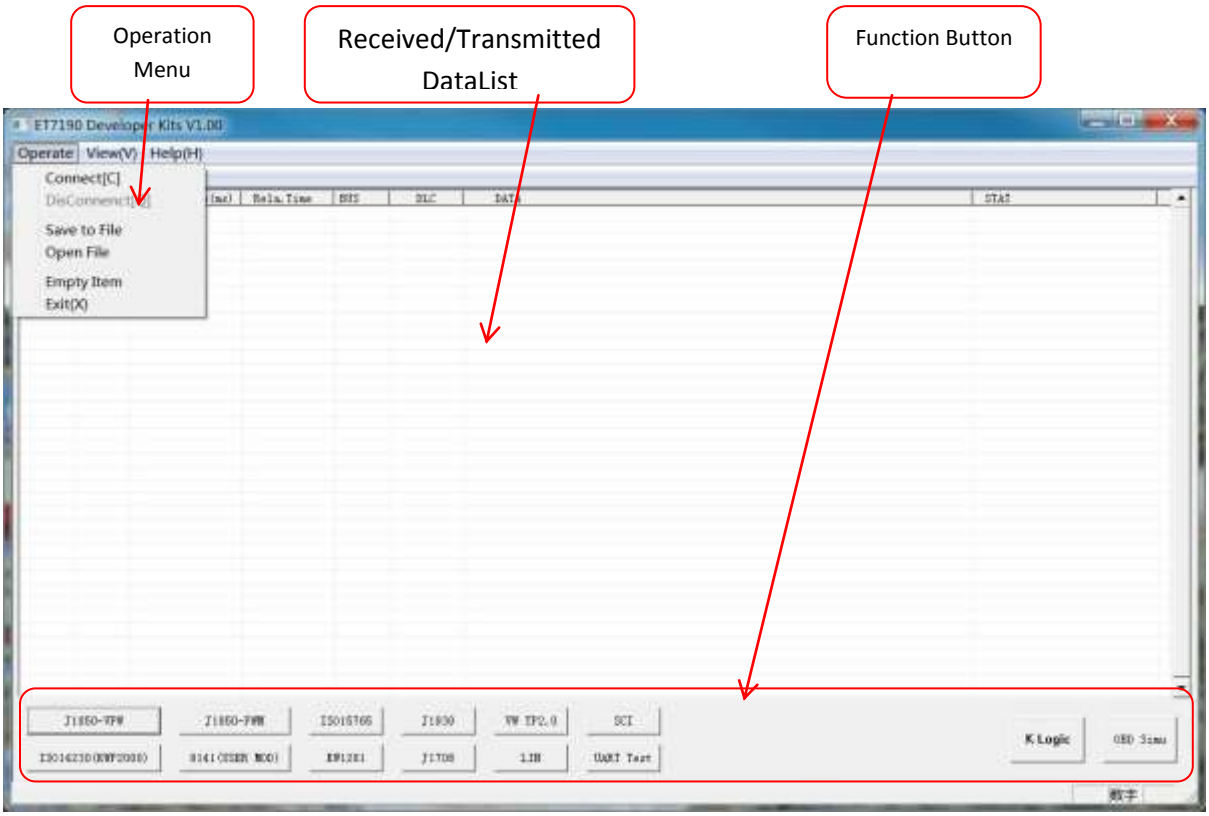
- Files under USBDrive catalogue are Windows (all versions are universal) USB drive and API function for USB operation provided by Microchip. In this demoboard, PIC18F4550 internal program is slightly modified and inserted in accordance with Microchip standard codes. In consideration of the length, USB programming is not furthered here, because a lot of examples and descriptions related to USB programming are available on MICROCHIP website.
- ET7190kits.exe application software.
- In simulating ECU opening, Obd2Default.DAT is default to load the respond data and setting. Any user-saved file (with the extension .SIM) of simulated data under UserFile Catalogue can be changed to Obd2Default.DAT to replace the original file and change the default loaded respond data and setting.
- API dynamic function library for USB operation is provided by Mpusbapi.dll Microchip.
- Userfile Catalogue: Saved user data

File extension: .rec files are the data log files received or sent in the main window saved by user.
.klg files are the K-line logic analysis files saved by user.
.sim files are the user-defined respond data files of ECU simulation saved by user.

2.2.2 Software Installation and Running

- Copy files under ET7190Kits Catalogue to the hard disk.
- Install USB drive: Insert USB device, find new hardware, manually install drive, select USB DRIVE Catalogue as the drive catalogue to complete the installation.
- Run ET7190Kits.exe.

2.3 Window Interface of ET7190KITS Tool



- **Monitor List of Received or Sent Data in the Main Window:**

All input or output data, or bus data, bus error information under monitor will be displayed in this list, in which the displayed time is WINDOWS system time that sends or reads

ET7190 buffer instead of the exact real time. Double click the list items to view more information and copy data in this list.

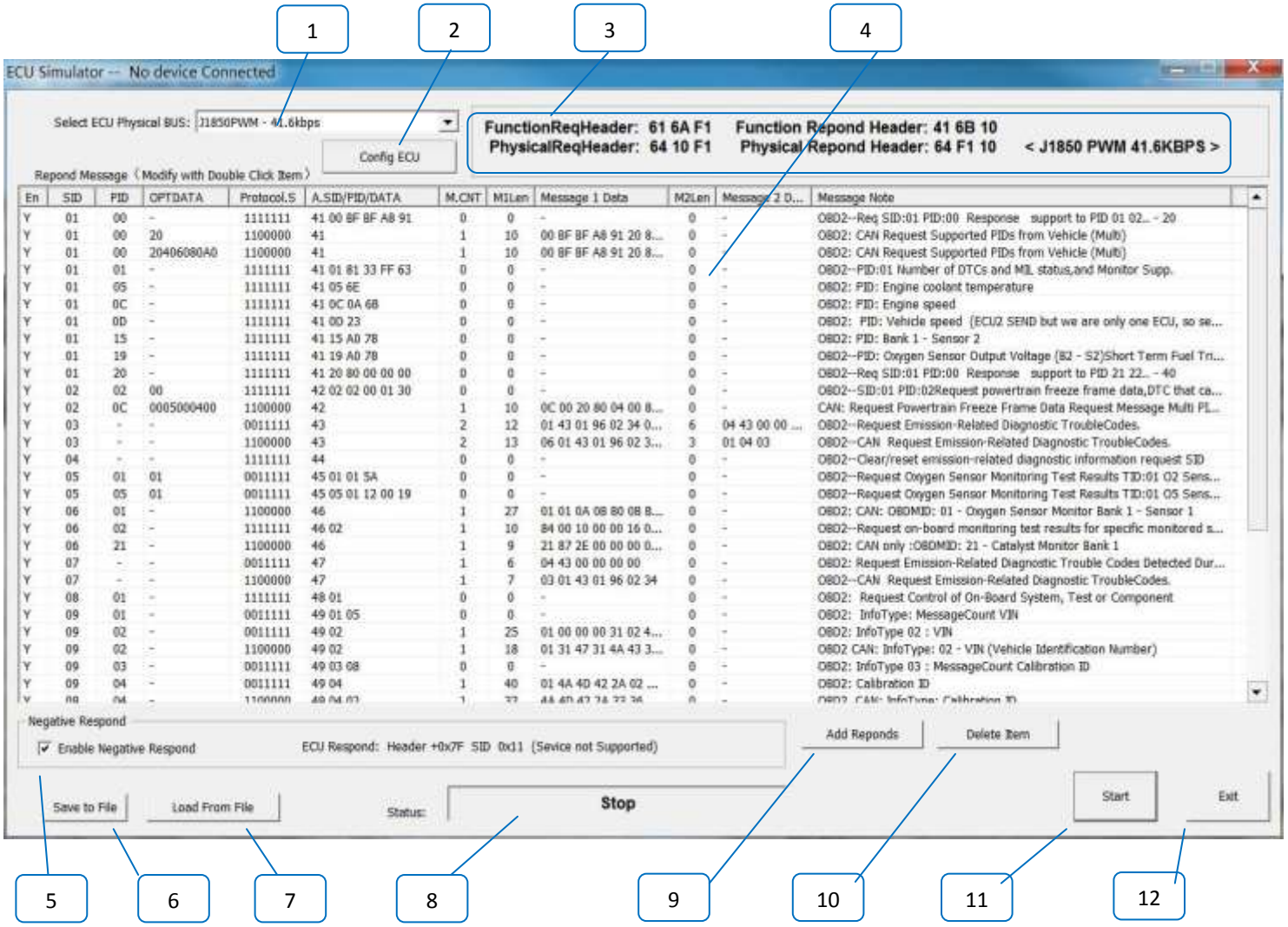
- **Operation Menu:**
  Connect: Select a device to connect before entering the corresponding functional operation.
  Save to file: Data list can be saved into file.
  Open file: Write in file data to the list.
  Clear data items: Clear data items in the list.
- **Function Button**: Enter the corresponding test function.

2.4 ECU OBD Diagnostic Simulation

2.4.1 Function

Full-functional OBD for the diagnosis of ECU simulation function. Any application protocol of ISO9141-2/ISO14230(KWP2000) /J1850VPW/J1850PWM /ISO15765 can be simulated. The initialization process and respond data are fully user-defined, every module on the vehicle can be simulated, when ECU is simulated on ET7190 demoboard, work timing is accurate and normative.(Except for CAN multi-frame output, only display the first frame information of sending FF)

2.4.2 Interface and Description



1. Select the physical bus in current simulation: Bus Baud rate may be changed in

configuring ECU parameters.

2. Configure ECU parameters: Message header (ID) of ECU request, bus Baudrate, sending time related to K-line protocol can be changed in simulation.

3. Key information in simulation of current protocol. In simulation ECU may define the ID of a function request and the ID of a physical address request respectively. ECU normally responds to the request of these two methods.

4. Define the respond data, user may modify, delete or add these data.

5. Upon the receipt of a request, if no suitable respond is found in the defined respond data, then negative respond will be automatically sent. (Service not supported), if this option is not ticked, there will be no respond. For development of application software, user should pay attention to how to deal with negative respond and no respond, which are found in actual application.

6. Define the modified respond data and save the defined ECU parameters to user files for the convenience of user.

7. Load user-defined data files.

8. Prompt the current status.

9. Add an item of user-defined respond data. (Double click the list items to modify or add data)

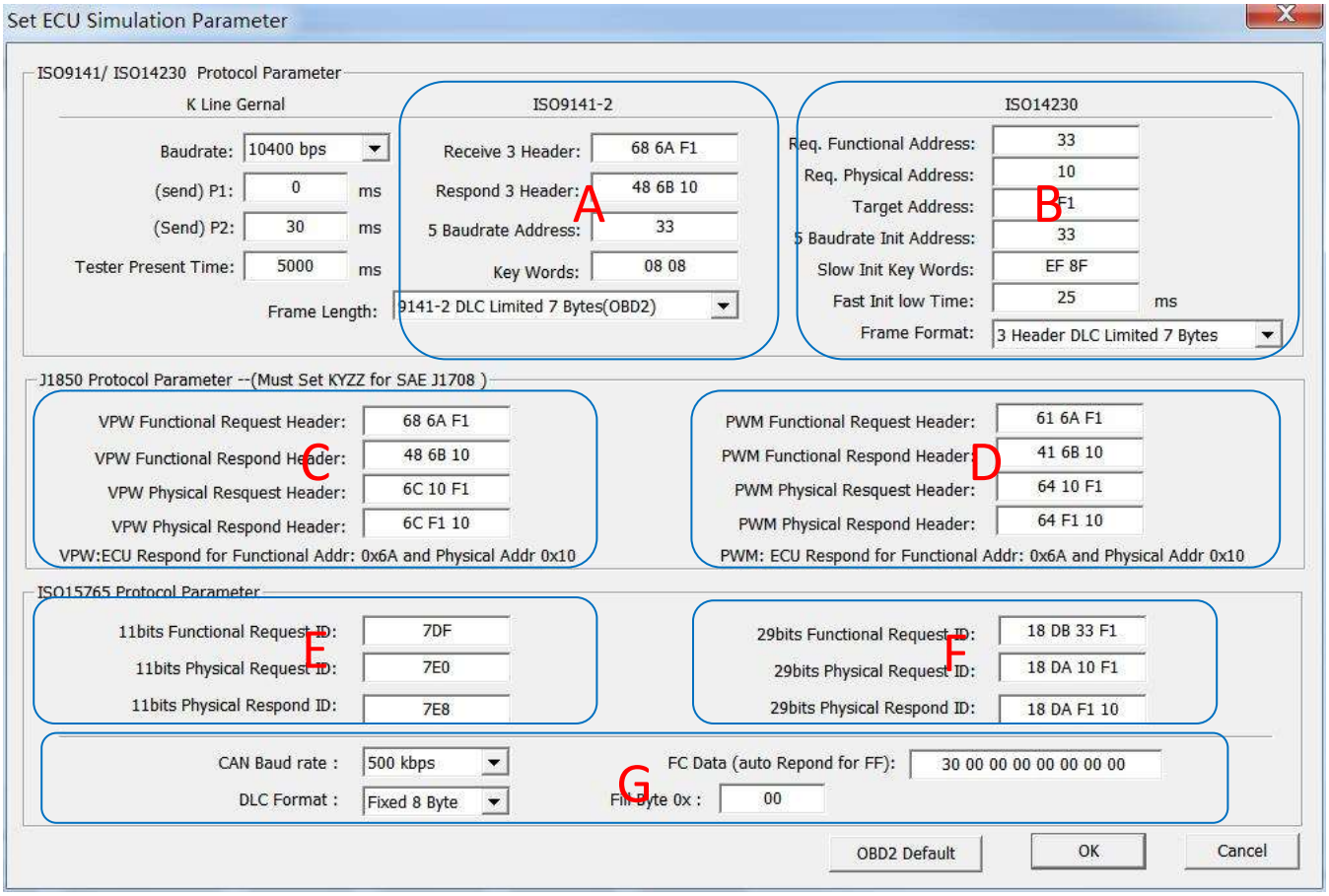
10. Delete one or more items of respond data. Press CTRL or SHIFT button+Click more items to delete.

11. Start or stop bus simulation, start running. User-defined data must not be modified in running.

12. Exit simulation interface.

2.4.3 Configure ECU Parameters

The following ECU parameter settings should be saved when saving simulation data files.



1. K-line Parameter Configuration

- K-line Baudrate, bus Baudrate at work, OBD2 and common definition is 10,400, but some ECU may also use 9,600, where simulation is variable.
- Interframe sent byte space is P1, ECU is generally 0ms
- Sent interframe space. For OBD2, it must set in 25-50ms, when simultaneously connecting several demoboards to simulate several ECU work, it must set different values with 1ms slot. For any application with length byte like ISO14230-3, it may set as 0ms, when multiple frames may be continuously sent.
- Maximum connection hold time, if requester exceeds this time with no request sent, the connection of simulation program will be terminated, when reinitialization is required.
- As shown above, Block A is for the definition of parameters related to ISO9141-2: Address, keyword, request protocol byte header, and respond protocol byte header in case of initialization. This initialization address, protocol header and related data may be changed to simulate other ECU modules of ISO9141-2 protocol.
  When ISO9141 is applied for the diagnosis of OBD2, the length of force frame is 7 bytes to the maximum. In simulating ECU, long message with over 7 bytes will be automatically

sent frame by frame. In case of no limit, then long message will be sent in one frame, which can deal with the simulation of special ECU module.

- As shown above, Block B is the area for the setting of ISO14230.

  In simulation of ISO14230, ECU will respond to functional address request (0b11xxxxxx xx F1) or physical address request (0b10xxxxxx xx F1), generally, these 2 requests are simultaneously supported in actual vehicle engine module. User can change the functional address and physical address in simulation and simulate ABS/SRS with other ECU. In simulation of this program, data frame request with additional length bytes are simultaneously supported.
  Default settings are OBD2 and physical address in 0x10 engine module.

  In ECU respond, ISO14230 frame sending length may provide 3 options below:

  A. 3 bytes header, with no additional length byte, maximum 7 bytes, as mandatorily required by OBD2. Respond long message will be automatically sent frame by frame.
  B. 3 bytes header, with no additional length byte, with no length limit. Long message is automatically sent in one frame, but when message length >63 bytes, length byte will be automatically added.
  C. 3 bytes header, always with additional length byte, with no length limit. Long message is automatically sent in one frame.

- As shown above, C/D are the areas for the setting of J1850VPW/J1850PWM respectively

  In setting J1850 protocol, the first byte of protocol header is generally the above set value subject to no change. For any need of change, J1708 protocol must be fulfilled, this simulation program only supports the format in 3 bytes header, and it writes in no simulation of 1 BYTE protocol header.
  In protocol header, the respond physical address must be identical to the request physical address.
- As shown above, E/F/G are the areas for the definition of CAN

  The setting principle is the same, for different ECU modules, only change ID and respond data. The default setting is the definition of OBD2 standard.

2.4.4 Add or Modify Respond Data

Click “Add Respond Data” button in simulation interface, or double click one in ItemList to

open the interface of editing respond data, add or modify respond data.

1. Current respond data enable: It only works in enabled status, for the same request service, multiple respond data items may be set, the respond data may vary from data item to data item, and any unused item may be disabled. Different data can be quickly switched to carry out the respond experiment.

2. Request service SID and data, ECU will respond to this message following the receipt of request data of this service (the length and data must be identical).

3. Respond data-supported bus: Because OBD2 slightly differs from long message multi-frame data and some data frame, CAN and other bus, this respond data-supported bus must be selected to define data. If data definition is identical, then tick all.

   For the reading of 03 TroubleCode request as previously described, CAN respond data frame appears one more byte of current TroubleCodes, so we shall define two data items for 03 SID service in respond data, one supports CAN data item and the other supports data item of other protocol.
   In starting the corresponding bus, the data item of corresponding protocol should be

automatically select to respond.

1. Note to current respond item for the convenience of view. No more than 128 characters are allowed to type in.
2. Respond service ID, if it is only single-frame respond in length <7 bytes, all respond data can be directly informed. Message1/Message2 behind it must be null (no data). If multiple frames require respond sending (in the length＞7 bytes), only type in the bytes identical to SID in each frame to ACK.SID.
   For example, 0902 Respond Data is given below (Non-CAN):
   49 02 01 00 00 00 31
   49 02 02 47 31 4A 43
   49 02 03 35 34 34 34
   49 02 04 52 37 32 35
   49 02 05 32 33 36 37
   Type in the first two bytes 49 02 to ACK.SID/PID/DATA for responding to request 09 02 service, and type in Message1:
   01 00 00 00 31
   02 47 31 4A 43
   03 35 34 34 34
   04 52 37 32 35
   05 32 33 36 37
   In simulation program of frame-by-frame output, automatically add ACK.SID/PID service byte in front of each frame.
3. Sent FrameLen in each frame: This item is invalid for CAN (CAN frame-to-frame sending is automatically processed as per ISO15765-2). In simulation program, when the FrameLen of Message1/2+ACK.SID/PID is bigger than the FrameLen of frame-to-frame sending, it will be automatically sent frame by frame, actually data number in each frame sent from Message1/2 is the number of bytes of FrameLen - ACK.SID/PID.
   The maximum FrameLen mandatorily defined by OBD2 is 7 bytes.
   This option is used to change the FrameLen of each frame only for the simulation of special module. It is also valid when ECU parameter is set as ISO14230/9141 limit length or J1850 protocol.
4. Message 1, maximum 256 bytes. In simulation program, when the FrameLen of Message1/2+ACK.SID/PID is bigger than the FrameLen of frame-to-frame sending, it will be automatically sent frame by frame. Simulation program can be set to make 2 message responds to a request service. (Each message is either single-frame or multi-frame message)

Message 2, the same as above. In setting Message 2, Message 1 must contain data, or it might not respond to the content in Message 2.

5. Import message from monitor list in main window.

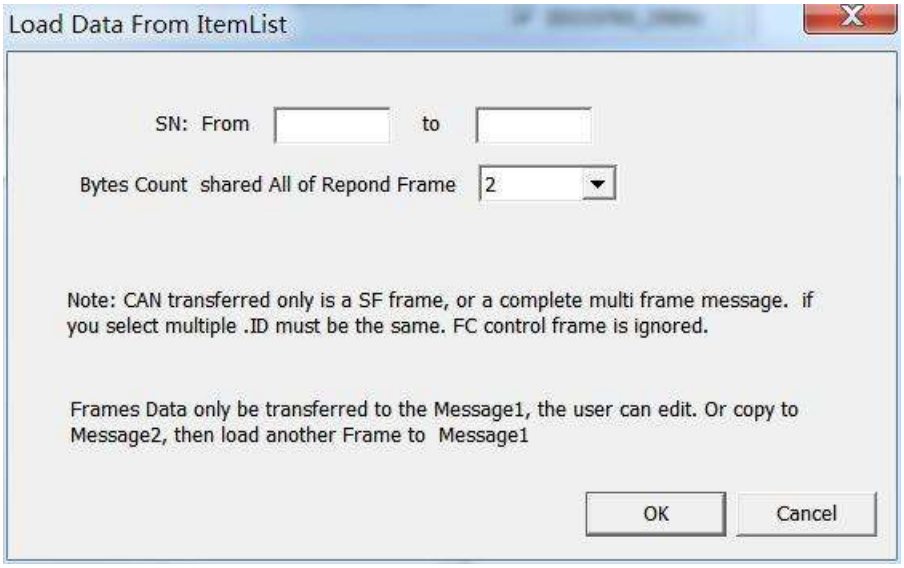
This function can be used to easily set respond data.
For unknown diagnostic protocol, user may use the monitor function of demoboard, (Filter is configured well to only receive the diagnosis-related ID) When the original diagnostic tester works, communication data on bus between the diagnostic tester and ECU can be completely

recorded in the monitor window. Data log will be saved after being completed. User is permitted to set the simulated data of ECU. Directly set by loading monitor data to respond items.

As shown below:

Type in the range of serial number of data log. (In case of only one frame, just type in the previous serial number, and the next one must be left blank)

Type in the length of service ID in each frame (CAN long message is irrelevant to this item).



User may load Example.rec data log files, where contains J1850/CAN/UART/ single-frame, multi-frame message logs. Load log item. Load method is understood through multiple tests.

2.4.5 User-defined Simulation Example 1: J1850 GM Airbag Module

Illustrating a simple example, setting and other special ECU functions are understood and simulated via J1850 in GM SRS module.

1. First, we set ECU parameter to enter Configure ECU Parameters as described in Section 2.4.3. Set VPW protocol header in Block C. The physical address in SRS module is 0x58, function request is not supported, but we enable such function request, which will not affect its response to physical address.

   Functional Address Respond Header is changed to: 48 6B 58

   Physical Address Respond Header is changed to: 6C 58 F1

   Physical Address Respond Header is changed to: 6C F1 58

   Save setting to complete the change of ECU parameters. (Only support J1850 VPW. Other

parameters are irrelevant with no need to change)

2. Add respond data: (Data is recorded from tester when it works)

First, we delete all available respond data, (after selecting multiple items of data in simulation window, click delete button). Then add respond data. We only add the following 4 service requests:

A. SID: 14 Respond: 54 Function: Clear TroubleCode
B. SID: 18 00 FF 00 Respond：Multi-frame TroubleCode Function: Read TroubleCode
C. SID: 20 Respond：60 Function: Enter the system
D. SID: 2A 13 06 FF FF FF Respond：ECU Identified Information Function： Read ECU identified information
Note: In responding to TroubleCodes, each frame only responds to a TroubleCode, so FrameLen of each frame only has 4 bytes in respond setting.

3. After informing the respond data, we will select J1850 VPW protocol from Option 1 in the interface as described in Section 2.4.2, and save as file (user-defined file name, import this file in the next simulation of this ECU, just directly start this simulation)
4. Data has been setup, click “Start Simulation” after connecting well to 1850 VPW bus, GND, and BATT of diagnostic tester. In this case, diagnostic tester can read TroubleCodes, clear TroubleCodes, and read ECU ID.
5. Under USERFILE Catalogue, J1850VPW_GM_SRC_6C58F1,SIM file is this example, which can be directly loaded prior to the simulation test.

Note: Add the respond data of SID service, whichever is first, after inserting data, automatically sorting SID by service ID. View the existing SID respond setting to understand the method of data setting.

2.4.6 Example User-defined Simulation 2:

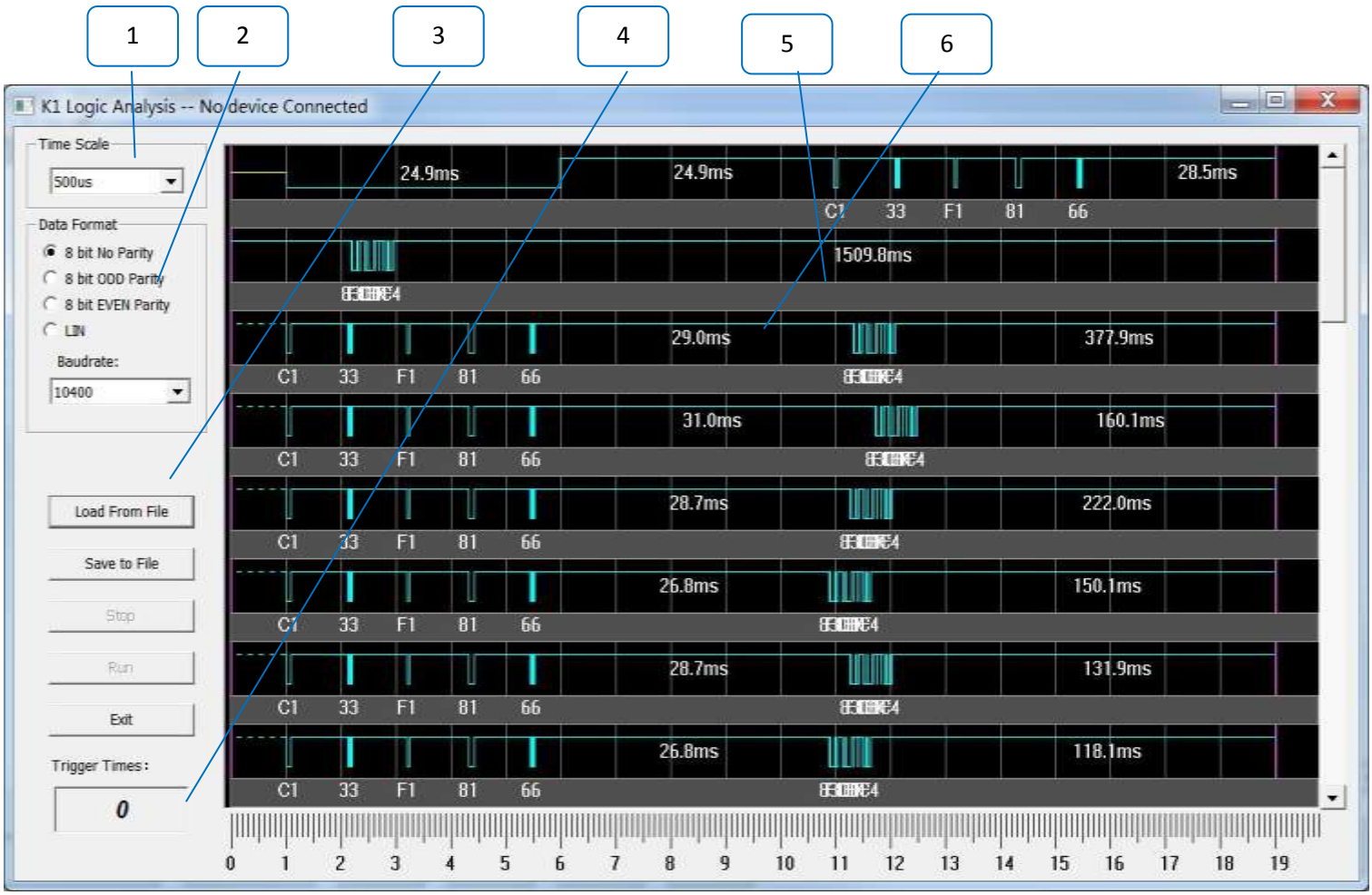
To Be Continued

2.5 K-line Logic Analysis

K-line has been extensively used in vehicle diagnosis due to low unit price of communication node, however, specific diagnosis has no mandatory standard except for OBD2 function, and there is no general standard for manufacturer in the initialization process prior to the communication, the vehicle bus analyzer is unable to record the triggering process in the process of initialization. Previously, logic analyzer was the only alternative, but its communication with vehicle enabled by function of data analysis was not so user-friendly because of complex usage and limited storage. This software is a good solution to this problem at nearly zero cost. And it is user-friend with no data leakage. Theoretically, it allows for infinite records. All times are visible and clear in K-line working, which gives a lot of help for user to understand ISO14230, ISO9141, LIN, J1708 or other K-line work timing.

2.5.1 Logic Analysis Interface

The logic waveform of ISO14230 quick initialization process is given below:



1. Time scale, which corresponds to the time of every grid in the bottom of window. Time scale is subject to change from time to time.

2. Selection of data analysis format: (Irrelevant to acquired waveform) After acquiring the figure, analytic test of all Baudrates and data formats can be selected, if incorrectly selected, data in the figure will be tailed with "#". View data with the appropriate Baudrates.

3. Command button:

4. After starting the acquisition, the number of level change in actual acquisition will be displayed, this software limits up to 100,000 times.

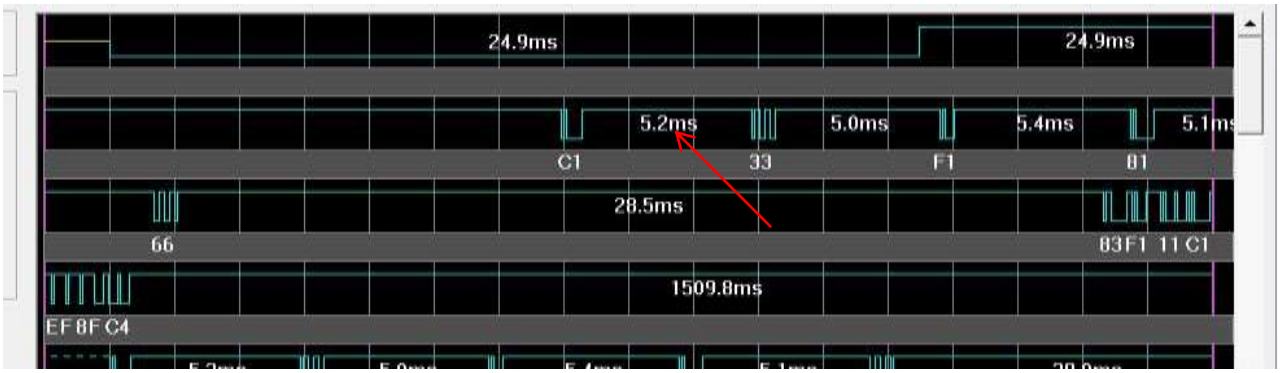
5. Table of logic waveform:

It can be clearly seen from this figure that K-line sends out the command of establishing the connection of C1 33 F1 81 66 after a 25ms low level and 25ms high level, after a delay of 28.5ms, ECU begins to respond to data frame.

It also suggests that byte slot of command sent by diagnostic tester is about 5ms.

6. In case of high, low level, for the slot of transfer time bigger than 1 byte, this software will display its time value, rather than the real slot of time delay. It is real time of high level. Real time delay slot must subtract the last continuous high-level time of the previous byte sent+One stop bit time. If the above figure is magnified:

   This diagnostic tester sends 0xc1 and 0X33 with the time slot of 5.2ms, because the previous byte is C1 and there are 2 high levels+1 stop bit, so the real byte slot must be 5.2ms - 3 bit time, or about 5.0ms.



2.5.2 Data Acquisition Process

- Connect diagnostic tester to vehicle (we assume that tester diagnoses via K-line), connect 3 lines in the demoboard, K1/BATT/GND, to vehicle. K1 is the K-line that connects to vehicle.
- Click "Start" button to start data acquisition.
- The diagnostic tester begins to diagnose, logic waveform and data in diagnosis will be directly displayed in computer. If data is incorrectly displayed, try to change the data format

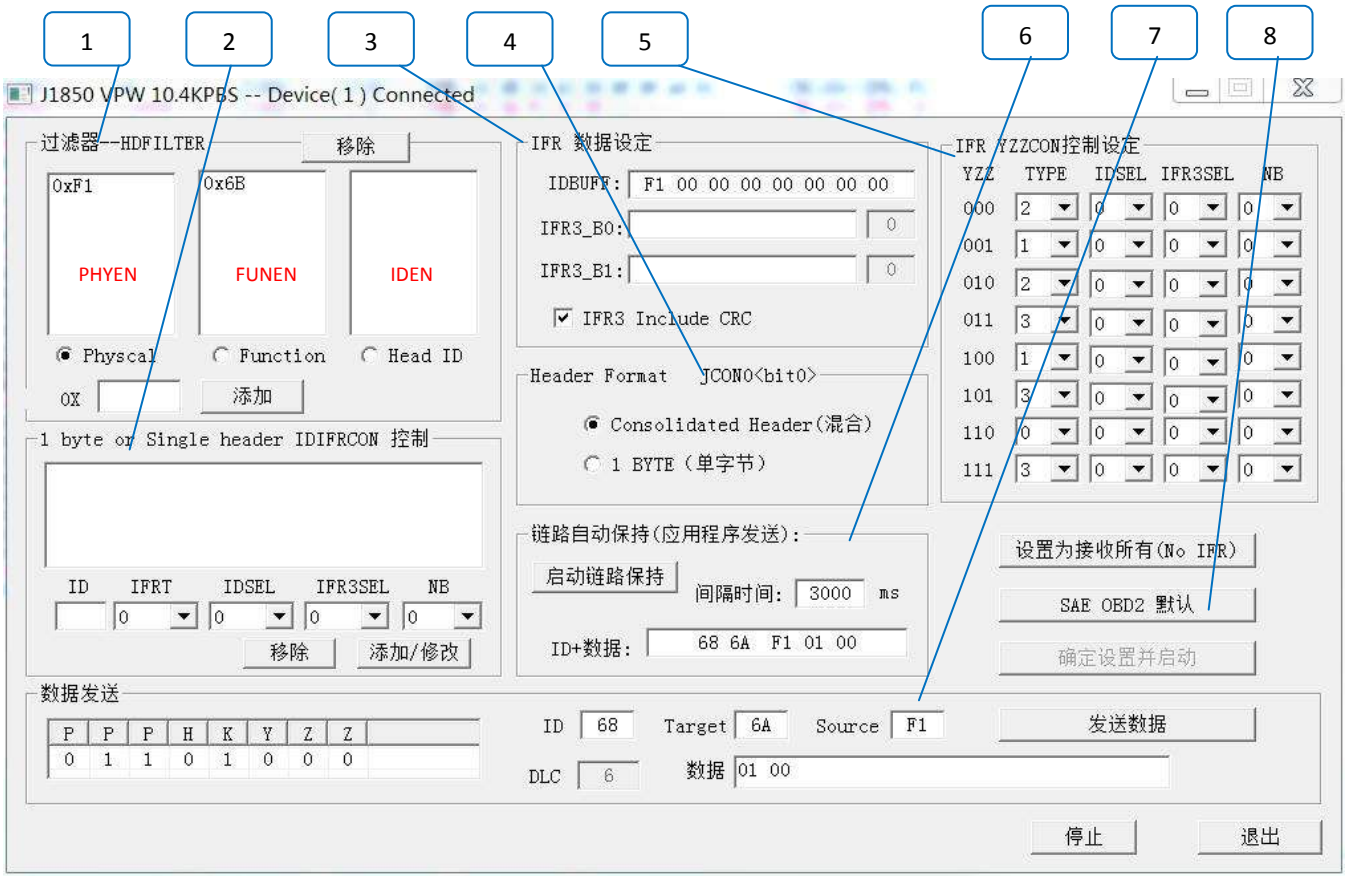
until the most appropriate is found.

- Acquisition may be stopped at any time. After stopping, save as file (Inform a suitable file name for the convenience of view), several simple example waveforms are gathered under Userfile Catalogue.
- Generally, we just collect the early stage of diagnosis by diagnostic tester and the waveform of several data frames. Through the acquired logic waveform we can know the Baudrate of bus, initialization process and protocol status. We can monitor and record a full set of data via relevant protocol in development tool. Based on monitor data ECU data can be defined and simulated for the purpose of ECU simulation.

2.6 J1850 VPW/PWM Functional Test

2.6.1 Description of User Interface

J1850VPW and J1850PWM are identical in terms of interface operation, but NB bit PWM in the interface is irrelevant, NB bit only exist in VPW bus, PWM is invalid.



1. Filter setting of HDFILTER register set: Correspond to functional address, physical address and ID filter of 1 BYTE header data frame respectively. Generally set physical address F1 and functional address 6B for the setting of diagnosis.
2. IFR respond control in 1 BYTE header, generally, with no need to set up. This function only enables user to carry out 1 BYTE IFR1/2/3 respond test.
3. IFR interframe respond data setting, generally, IFR1/IFR2 respond value is the physical address F1 of this device, so only need to set IDBUFF[0]=0xF1, generally IFR3 is not required to set, where the interface function mainly permits user to carry out IFR1, 2, 3 respond experiment between several demoboards.
4. Selection of frame data format, generally Consolidated Header is selected. And generally H bit of protocol header is always 0. (3-byte header)
5. YZZCON control of Consolidated Header, generally SAE J2178 OBD2 is default setting, but if bus line is only monitored, then all YZZCON.IFRT must be set as 0, or this module may automatically respond IFR1/2/3 in accordance with the setting in receiving data, which will affect the bus data.
6. Connection hold, no connection hold is needed for the diagnosis of J1850, if required, manual starting is allowed and connection data frame is automatically set.
7. Sent data frame, ID and data may be modified by user.
8. Several functions are classified by:

"**Set as Receive All**" Set as receive all data, YZZCON IFRT are all 0, no respond will be made in receiving data. This is identical to monitoring the status of bus line data. HDFILTER or filter is set as receive all functional addresses, physical address and single ID frame. To clearly monitor and diagnose relevant contents, all addresses and IDs in the filter irrelevant to diagnosis must be deleted or removed.

"**Set OBD2 Default**" Configure for the diagnosis of OBD2 as per J2178 standard.

Any modification of configuration data is only valid after stopping and restarting bus line. After starting the bus line, any change of configuration is only made on the computer interface, which doesn't write into ET7190 chip. It only writes in when starting the bus line.

2.6.2 J1850 Bus Data Monitor

1. Connect Bus VPW: Connect BATT/GND/VPW in parallel to vehicle and diagnostic tester.
   Connect Bus PWM: Connect two lines PWM+ PWM-, other lines may not be connected.
2. Set the filter based on needs.
3. All YZZCON.IFRT should be set as 0, with no IFR respond.
4. Start the bus line. Any required data in the bus line will be received.

2.6.3 Diagnosis Request Data Transmission

Take VPW for example (no difference with using PWM):

1. Connect to vehicle bus and connect to device, enter VPW interface, click "SAE OBD2 Default Button", and click "Confirm Device and Start". After that, data transfer is ready.
2. In the diagnosis of OBD2, just inform the data to be sent in the sending data item, click "Send Data" button, then data frame will be sent to bus line. ECU will rely respond frame.
3. Diagnosis of other modules: For example, the simulated SRS diagnosis as previously described in Section 2.4.5.
   Just change ID/Target/Source in the sending area to 6C 58 F1(Header)
   After informing in the data area: 19 02 FF 00, click "Send Data" button. Send to SRS the 1902FF00 request command of reading TroubleCodes. (Filter is not required to change, generally request and respond in physical address mode like other modules, target address sent from ECU is F1)

Note: In the diagnosis of VPW, IFR respond is not required, but we set YZZCON in the value identical to PWM. Because VPW request ID is 6C or 68 (k=1), with no IFR respond; and PWM request ID is 61 or 64 (K=0), with IFR respond, so there will be no confusion.

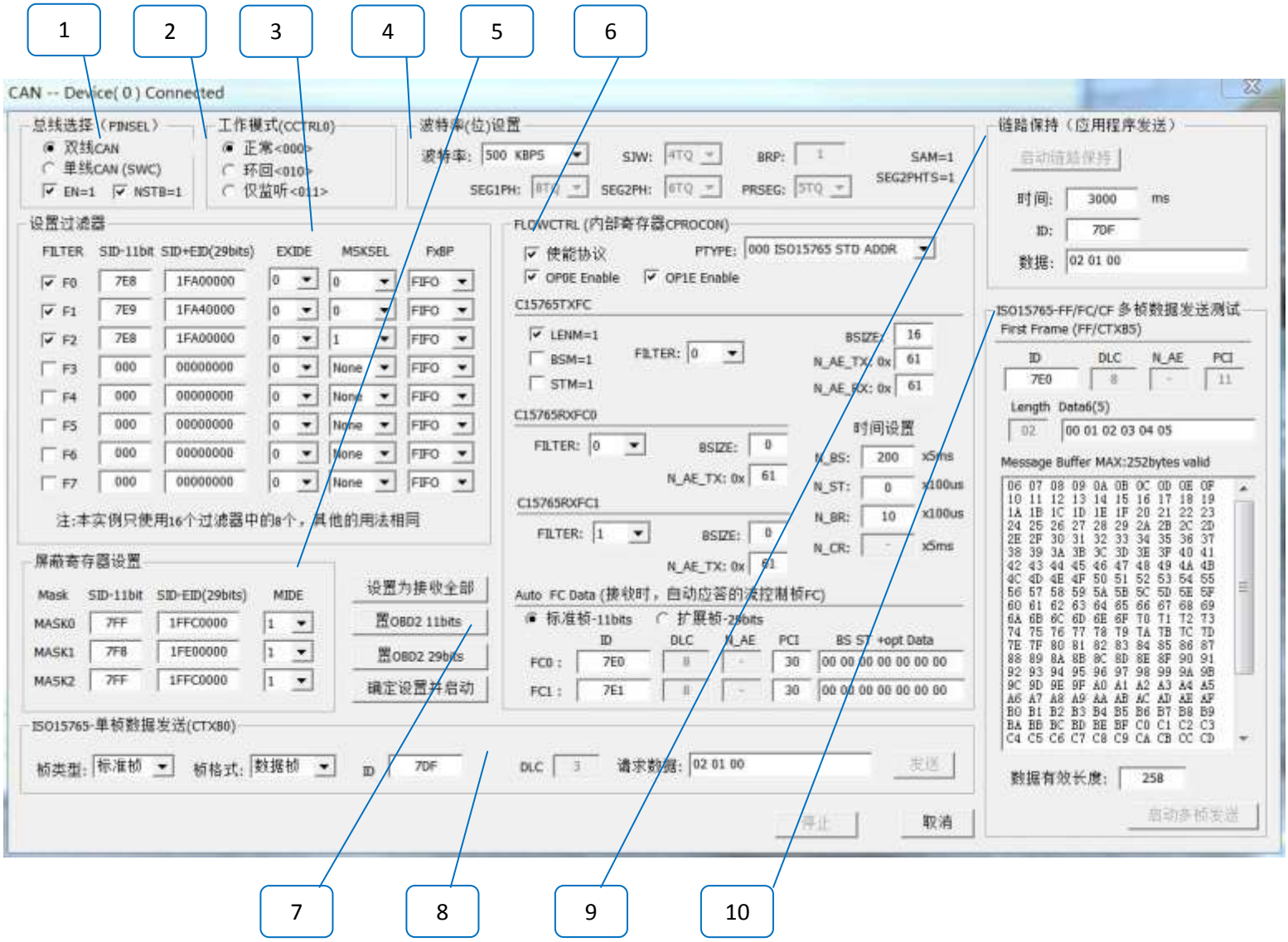
4. Connect two pieces of demoboards, one simulates ECU work in J1850 bus, which can diagnose the respond test of command.

2.7 ISO15765 Functional Test (With Single-line SWC)

ISO15765 functional test includes double-line CAN and single-line CAN. 11-bit standard frame executes the communication in ISO15765 Standard Addr mode and Extend Addr mode. And 29-bit extended frame executes all functional tests of communication in ISO15765 Standard Addr mode and Extend Addr mode.

User can simulate the communication between two pieces of demoboards, in order to fully understand the functions of ET7190 register and register buffer (buffer area) in CAN communication.

2.7.1 Description of User Interface



1. Selection of bus: Double-line CAN or single-line CAN is optional.
   In single-line CAN, SWEN/NSTB option can control the operation mode of single-line CAN bus drive chip (AU5790, etc.), note that in single-line CAN, bus Baudrate must be below 100KB, GM normally works at 33.3KBPS and 83.3kbps. Hardware circuit of Single-line CAN is unable to support the communication rate above 100KBPS.
   If at 33.3KBPS, then it must have SWEN=1 NSTB=1 (normal operation mode)
   If at 83.3KBPS, then it must have SWEN=0 NSTB=1 (used in ECU refreshing)
   About GM3110 application layer protocol: (use 11-bit ID, double-line, single-line CAN are not differential)
   Two definitions are given to GM CAN network layer, namely UUDT and USDT.
   USDT Protocol, identical to ISO15765-2(Extend Addr and Standard Addr), is a divisible data transmission protocol.

UUDT Protocol, with no PCI byte, is a common CAN data frame, no divisible data transmission, always as a single frame respond. UUDT and USDT data simultaneously exist in bus line. The feature that differentiates two data as mentioned above is different ID definitions, partial ID finishes USDT transmission, while special ID is assigned for UUDT transmission.

2. CAN operation mode:

Loopback Mode: Only a piece of demoboard can be used for receiving or sending test, actually no data output is identified in bus line.

Listen-only Mode: Test bus Baudrate. If there is data in bus line, in case of incorrect Baudrate, CAN module can receive the error flag, under this mode, CAN will not send error frame. When OBD2 automatically searches protocol, CAN bus (250KBPS or 500KBPS) must be tested before it. Normally this mode is required to test whether the bus line can normally receive data or receive error information, and determine if Baudrate in current test is correct or not, in order not to affect the normal transmission of vehicle bus line.

Normal Mode: For normal communication. If Baudrate is incorrectly set, start bus line, CAN module will send error frame when receiving error data due to data transmission in bus line, which disables the communication in other modules. CAN bus doesn't allow for two different Baudrates set in the same bus.

3. Set filter:

Correspond to CRXF0-CRXF7 receive filter and CFMSK0/1, CFPNT0 filter buffer pointer, we only use 8 filters, CRXF8-CRXF15 is used in the same way.

4. Set bus Baudrate: Correspond to ET7190 CCFG0/1/2 register value.
5. Set mask register set: CRXMSK0/1/2
6. Flow control setting in ISO15765 multi-frame sending and receiving

ISO15765 long message data transmission is Node to Node transmission. ET7190 can automatically send long message frame-to-frame to a node. Meanwhile, it can automatically receive long message sent from two nodes, and automatically send one or more FC frames in receiving based on BS definition of FC in long message.

Flow control register: Default setting of CPROCON, C15765TXFC, C15765RXFC0, C15765RXFC1, CTXB6, and CTXB7 is the diagnosis setting of OBD2. For details, see bit definition of each register in ET7190 data manual.

7. Function button:

Set as Receive All: Set the filter as receive all data frames, including: 11-bit ID, 29-bit ID data frame and remote frame.

Set OBD2 as 11bits: Set all registers in 11-bit OBD2 diagnosis mode.

Set OBD2 as 29bits: Set all registers in 29-bit OBD2 diagnosis mode.

8. Content of data transmission:

For OBD2 diagnosis, in request data field, PCI byte must be informed, this program doesn't add any additional data, PCI byte constitutes 0x00|FrameLen, actually the first byte is FrameLen, e.g.:

Read TroubleCodes: 03 Input: 01 03

Read speed: 01 0C Input: 02 01 03

9. Connection hold:
Normally, CAN application doesn't require connection hold, which might be needed in special diagnosis, data frame content of connection hold can be set, and manually start connection hold,

10. Send a long message test.
Note: ID setting, node-to-node transmission, received ID must match with the sender.
OBD2 11-bit definition: Tester sent 7E0 Corresponding ECU ID: 7E8
OBD2 29-bit definition: Tester sent 18 DA 10 F1 Corresponding ECU ID: 18 DA F1 10

A long message must be successfully sent, ISO15765 flow control register and filter related to sender must be correctly set, or error setting will lead to the transmission failure.

2.7.2 Monitoring on CAN Data Bus

1. Connect CAN bus, if it is single-line CAN, connect SWC/GND/BATT to vehicle, if it is double-line CAN, just need to connect CANH/CANL.
2. Set the filter based on needs, only receive ID data concerned. In monitoring, close "Enable Protocol" function in CPROCON, avoid automatically responding CF frame in receiving FF frame.
3. Start the bus. Any required data in the bus will be received.

Note: ET7190 handling capacity is above 20,000 frames/s, but demoboard receiving capacity is about 2,000 frames/s due to PIC18F4550 functional limit, which is sufficient to common application.

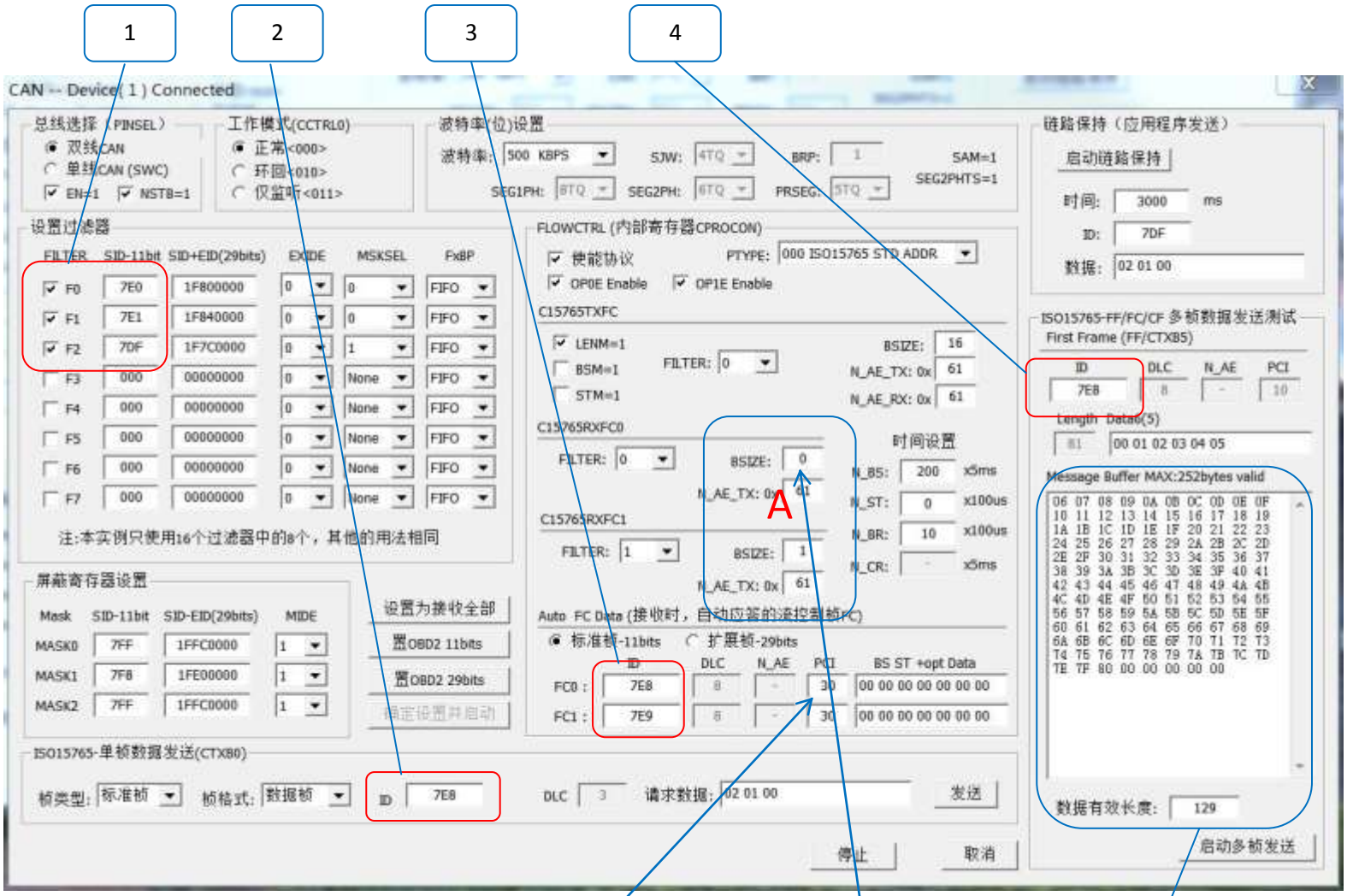
2.7.3 Long Message Automatic Frame-to-Frame Experiment

We use 2 demoboards to carry out the multi-frame sending experiment:

Node A, set as default OBD2 11-bit setting, send ID: 7E0 Received ID: 7E8

Node B, set as simulated ECU address, sent ID: 7E8 Received ID: 7E0

Node A load default setting, change no interface content, Node B is changed as follows:



1. Content changed to, filter set as receive tester address: 7E0/7E1/7DF
2. In sending one frame, ID changed to 7E8 (ECU →TESTER)
3. Automatically respond FC ID 7E8/7E9
4. In multi-frame sending, start FF frame send ID 7E8
5. After finishing setup, one frame and long message can mutually transmitted between Node A and Node B.

Note: FC frame data is set as BS=0 ST =0, so receiver will send all data in one go upon the receipt of a FC frame, if receiver requires multiple receiving, e.g.: BS=8, ST=5, receiver's FC data can be changed to 30 08 05 00 00 00 00 00, and receiver's C15765RXFC0.BSIZE=8

must be changed too (as shown above, at receiver's A)

Receiver doesn't need to make any change, if data is longer, after sending data frame in BS block size for each time, sender will suspend sending, upon the receipt of the next FC frame, it continues sending. In this process, multiple FC control frames will be received.

6. Long message setting:
   Because sent LENM is set as 1, after importing data into the buffer, add several 00 in the tail as stuffing byte, then rewrite the actual length of effective data. (The testing method is used only for this software)

2.7.4 Diagnosis Request Data Transmission

For OBD2 diagnosis, in request data field, PCI byte must be informed, this program doesn't add any additional data, PCI byte constitutes (0x00=PCI)|(FrameLen), actually the first byte is FrameLen, e.g.:

Read TroubleCode: 03 Input: 01 03
Read speed: 01 0C Input: 02 01 03

Connect two pieces of demoboards, one simulates ECU work in CAN bus, which can be used to diagnose the respond test of command.

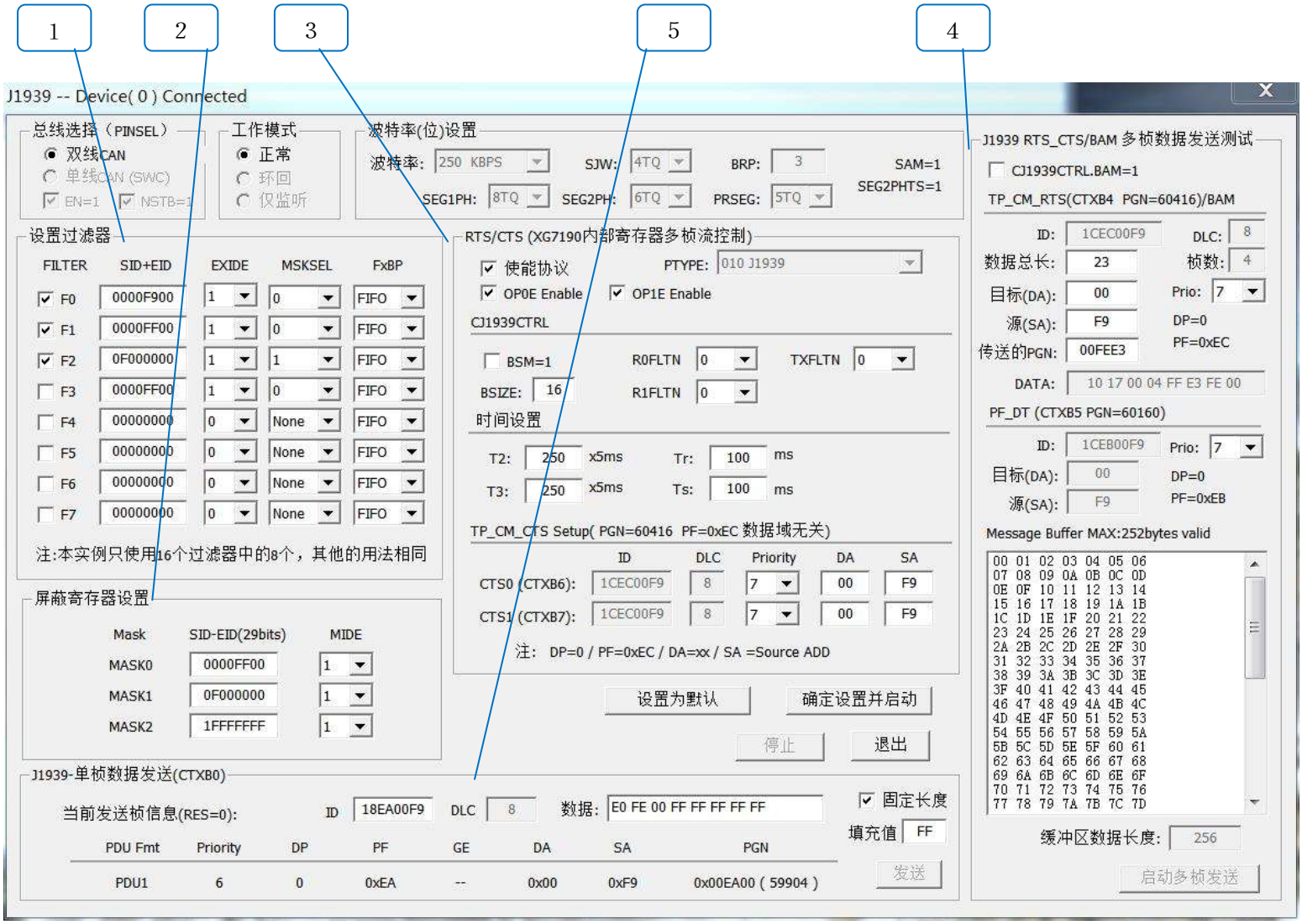
2.7.5 Notes for Single-line CAN

- Data transmission, receiving and filtering of single-line CAN are not different from those of double-line CAN.
- Single-line CAN Baudrate must not >100KBPS, GM only uses 2 kinds of Baudrates, namely 33.3 kpbs and 83.3 kbps.
  Single-line CAN bus drive sets the requirement for AU5790:
  If at 33.3KBPS, then it must have SWEN=1 NSTB=1 (Low speed)
  If at 83.3KBPS, then it must have SWEN=0 NSTB=1 (high speed)
- The above-mentioned communication test can be done between two pieces of demoboards with no difference from double-line CAN.

2.8 J1939 Functional Test

J1939 uses 29-bit ID, 250kbps Baudrate for communication, this function can enable J1939 one frame PGN request and send, long message automatic frame-to-frame sending, and automatic respond CTS control frame in receiving long message.

2.8.1 Description of User Interface



1. J1939 filter setting:

   J1939 filter setting is slightly different from ISO15765, with the general method as:

   - Filter 0, set as receive Target Addr. DA as the physical address of this module, when we assume module as tester, address 0xF9, combined with CRXMSK0, receive all DA=0xF9

data frames. EXIDE must=1, only receive 29-bit extended frame.

- Filter 1, set as receive global radio message, global radio message DA=0xFF, combined with CRXMSK0, receive all DA=0xFF data frames. EXIDE must=1, only receive 29-bit extended frame.
- Filter 2, set as receive all data frames of global parameter set PF=0xF0-0xF1 … 0xFF, CRXF2 combined with CRXMSK1.
- Other filters are out of use, for different modules, only need to change the Target Addr. Of CRXF0 filter, actually this address is Source Addr. of this device. Normally, Source Addr. of sent data frame is this value. Certainly, 16 other filters may receive other specific PGN data.
- Default set value of this interface: Source Addr.: 0xF9 Target Addr.: 0x00, as another node, only need to change Source Addr. of this device, different Target Addr. can be connected to any module in receiving data frame.

2. Setting of mask register: Combined with filter.

3. Setting of long message automatic frame-to-frame sending and automatic receiving.
   CPROCON setting:
   Enable Protocol, PTYPE is set as 010, J1939 support mode.
   OP0E/ OP1E enables CTS0 and CTS1 to auto-respond respectively, CTS respond can be made for RTS long message of two modules respectively.
   CJ1939CTRL setting:
   For BSM transmission in normal condition, BSM must be set as 0,
   For BAM dynamic change setting in Item 4, there is no option, if select BAM=1, send global radio long message, and Target Addr. of global radio message is always 0xFF.

   When BSIZE sends CTS, BS maximum must not be 0, set as 16 (If BSM=1, mandatory size, but it goes against J1939 standard).
   TXFLTN=R0FLTN=R1FLTN=0：Generally all set point at the filter CRXF0 of physical address (Node to Node) message in this module.

- Time setting: Ts /Tr can decrease time and accelerate sending, subject to adjustment based on real application.
- If CJ1939_T2 = 250 * 5ms receives multi-frame data, maximum awaiting time of DT frame.
  If CJ1939_T3 = 250 * 5ms sends multi-frame data, maximum awaiting time of receiver's CTS.
  If CJ1939_Ts = 100* 1ms continuously sends DT frame, time slot.
  If CJ1939_Tr = 100 * 1m receives multi-frame data, time delay of sending respond frame

CTS/EOM.

- CTS0, CTS1 frame setting
  For CTS frame, PF=0xEC, data length is 8 bytes at least, subject to no change, SA is the address for this module. DA and data field are not required to set, upon the receipt of RTS PGN1 request sent to this module, ET7190 will automatically set data field with DA for CTS0 respond. Before finishing PGN1 message, if another module sends a RTS PGN2 to this module, ET7190 will automatically use CTS1 to respond, and deal with PGN2 message.

4. Long Message Frame-to-Frame Sending
   - If CJ1939CTRL.BAM=1 is selected, message will be sent by global radio message.
     If CJ1939CTRL.BAM=1 is not selected, message will be sent by node-to-node RTS/CTS message flow format.
   - Setting of TP_CM_RTS frame (CTXB4 sending)
     RTS(BAM) corresponding PGN request is sent prior to sending long message at the source node. PF=0xEC(PGN60416) in this message is subject to no change, user informs Target Addr. and Source Addr. and PGN number to be transmitted. This program will automatically generate RTS frame based on the buffer data to be sent.
   - ID setting of PF_DT frame, this frame PF must be 0xEB, and its frame PGN=60160. Actually CTXB5 29-bit ID is automatically set as DT frame header based on RTS frame. When user programs, manual setting must be correct, or long message is unable to be sent.
   - Inform the data to be transmitted in data editing box, if the last frame is in insufficient length, inform FF in the tail.
   - Inform the FrameLen of actual sending in data total length edit box.
   - Click “Start Multi-frame Sending” to send long message.
   - In a complete process of receiving and sending, sender will receive one or more CTS frames, and an EOM message frame, after receiver receives one RTS frame, all data frames will be followed.

5. Send PGN request
   PF＝0xEA is one frame of requesting PDN data, when user informs PGN in data edit box, a PDN request will be sent.

   Modify ID, used to modify Target Addr. and Source Addr. Certainly, PF can be changed to carry out other functional tests. With ID modified, the following frame message will be automatically modified accordingly. For easy understanding, different IDs match with J1939 message format definition.

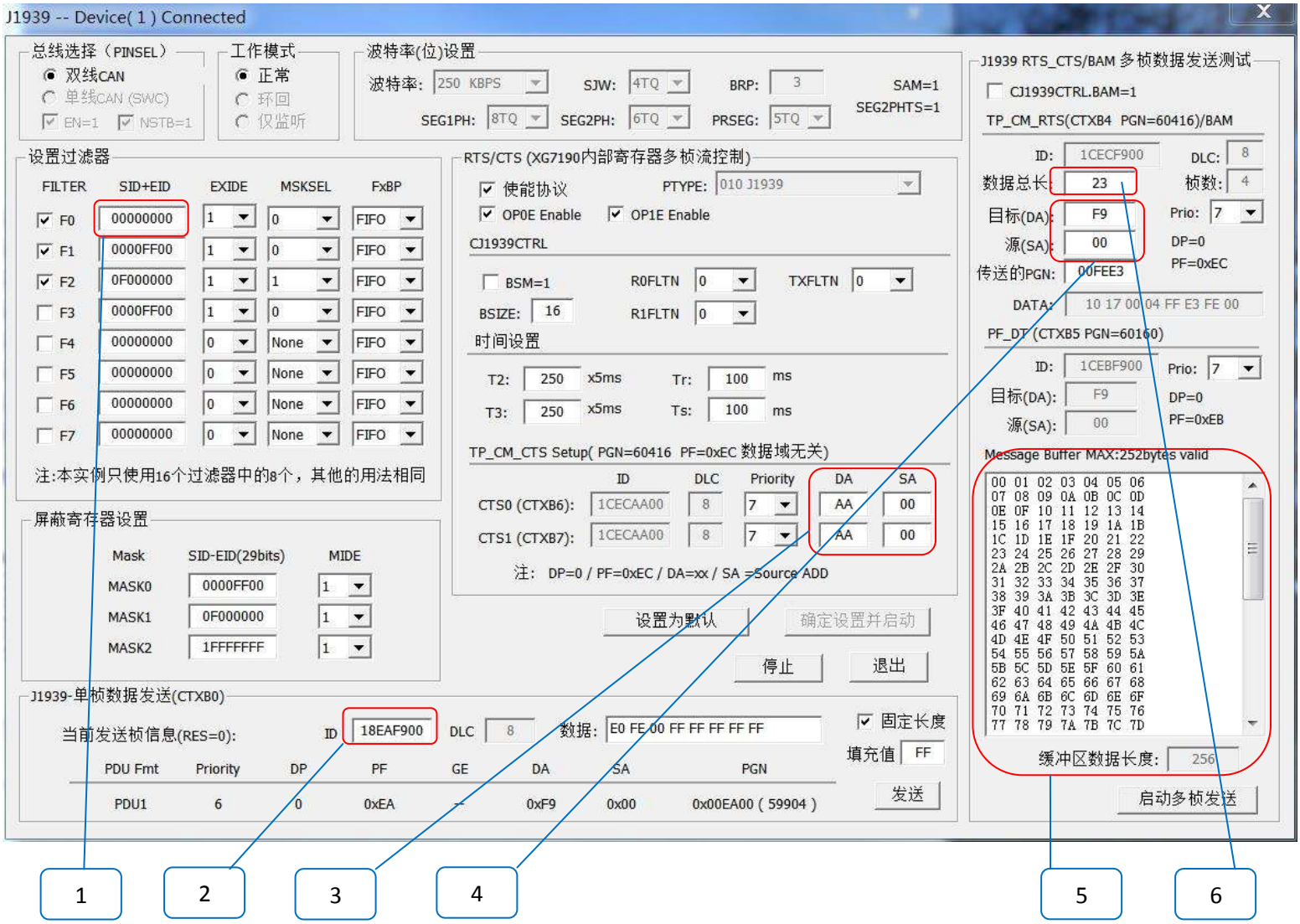
2.8.2 J1939 Data Monitoring

J1939 data monitoring is not different from common CAN, configure the filter, receive the number that you desire, but close CPROCON “Enable Protocol” in monitoring to avoid producing the automatic CTS respond upon the receipt of RTS.

2.8.3 J1939 Data Transmission Test between Two Demoboards

2 demoboards are used to carry out the receiving and sending experiment of one frame and long message:

Node A, set as default setting: Target Addr.: 00 Source Addr.: F9
Node B, Target Addr: F9 Source Addr.: 00
Change on Node B is as shown below:



1. Filter 0, set receive address changed to 0x00 (Source Addr. 00 in this application)
2. Send ID address exchanged from 18EA00F9→18EAF900
3. SA from F9→00, DA is any irrelevant value. (Source Addr. 00 in this application)
4. Source Addr. In RTS exchanged with Target Addr.
5. Inform data to be transmitted in data edit box, inform FF if the last frame is insufficient.
6. Inform the actual number of data to be transmitted. Press “Start Multi-frame Sending”, allowing for data transmission between nodes.

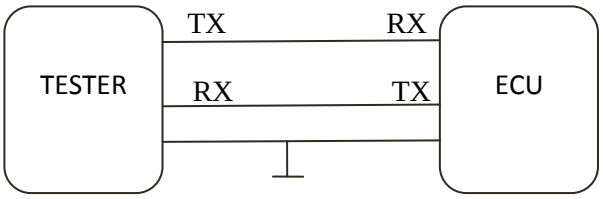
2.9 VW TP2.0 Functional Test

VW TP2.0 test requires to be combined with ECU.

This function is described until finishing VW TP2.0 ECU simulation function, Please Wait…

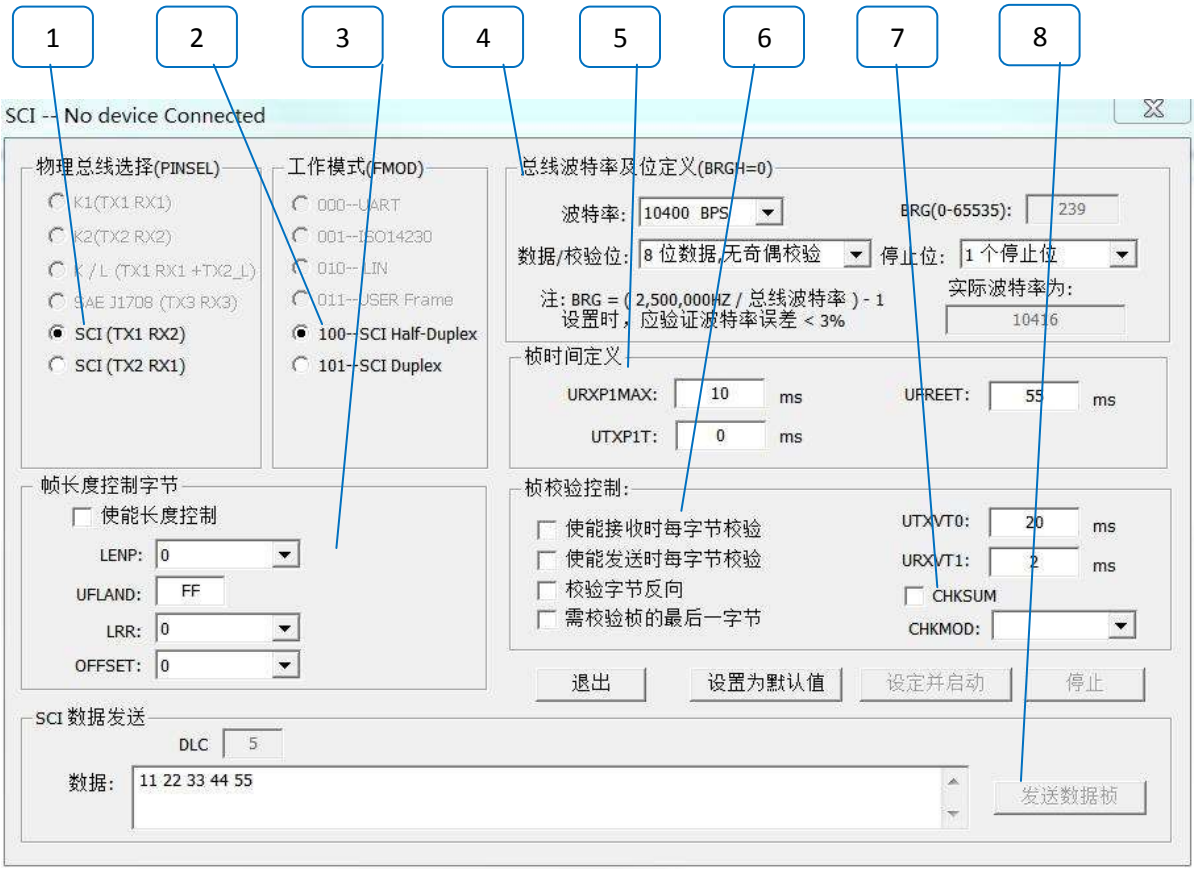
2.10 SCI Functional Test

There is no strict definition for SCI message format, which is generally user-defined based on real application. FrameLen, J2610 defines each frame with no more than 4,096 bytes, ET7190 maximum frame length of sending or receiving is 256 bytes in FMOD=100/101, for any frame more than 256 bytes, user needs to operate in FMOD=000(general UART mode).
SCI bus uses double-line communication, as shown below: One sending line and one receiving line. Note: When this demoboard uses SCI bus, logic level is BATT, if at 5V level, SI9241 power supply is required to change.



When PINSEL is configured as TX1, RX2 as the drive pin of EUART, K1 of demoboard is used as TX pin, K2 as RX pin; when PINSEL is configured as TX2, RX1 as the drive pin of EUART, K2 of demoboard is used as TX pin, K1 as RX pin.

2.10.1 Description of User Interface



1. Selection of bus: When PINSEL is configured as TX1, RX2 as the drive pin of EUART, K1 of demoboard is used as TX pin, K2 as RX pin; when PINSEL is configured as TX2, RX1 as the drive pin of EUART, K2 of demoboard is used as TX pin, K1 as RX pin.

   When two demoboards are tested, one is set as TX1/RX2, and the other is set as TX2/RX1 for SCI communication test.

2. Selection of SCI communication mode:

   In full duplex mode, byte check in Item 6 must not be done, when receiving and sending may proceed at the same time.

   In half duplex mode, receiving and sending must not proceed at the same time, only one-way data

transmission is allowed, byte check in Item 6 can be set.

3. In these two modes, the rule of length byte in a frame can be set.
4. Define bus Baudrate and bit.
5. Frame time setting.
6. Byte-verify setting, (see ET7190 data manual Section 8.7.2 SCI Message Transmission Method)
7. Verification and setting. In Enable Checksum, if the received frame CS bit is 0, it means check and accuracy, if 1, it means check and error of message frame, check of frame to be sent requires automatic calculation prior to sending with frame.

Enable Checksum has two algorithms, ISO14230/J1708, either of which serves as the common checksum algorithm.

8. Send data, if length byte is defined, the length byte in a frame must comply with your setting.

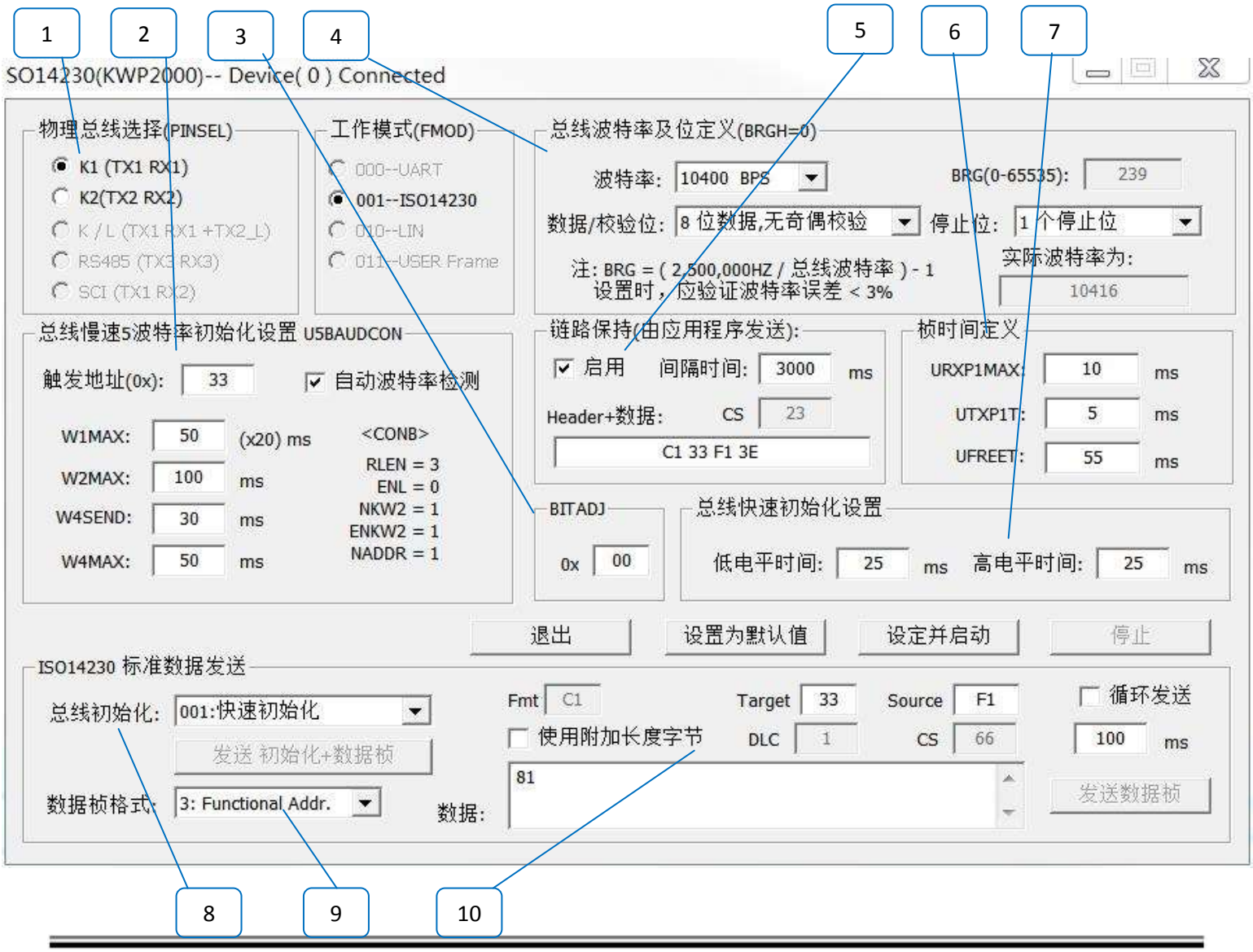
2.10.2 Test between Two Demoboards

According to the connections in Section 1.4 under Chapter 1, two demoboards are defined as different TX and RX prior to the sending and receiving test.

2.11 ISO14230 Functional Test

This functional test allows for ISO14230 quick or slow initialization, message header may use ISO14230-defined 3 formats for sending or receiving test, in order to fully understand the format, sending, receiving method of ISO14230 message.
Through user-defined Target Addr. and initialization method, it may communication with any ECU module in compliance with ISO14230.

2.11.1 Description of ISO14230 User Interface



1. Selection of physical bus

K1 and K2-line are identical, either one may be selected as the communication bus line.

2. In slow initialization, 5buad is used to trigger, trigger address 0x33 is 0x33 functional definition, trigger address may vary from ECU module to ECU module. K-line logic analysis function of this development tool can be used to obtain the initialization information from the original tester.

If slow initialization is used to trigger, it is recommended to use auto BaudRate, which may not be 10400 for different ECU modules, and 9600 BaudRate is also common. Auto BaudRate can be used for the automatic identification.

3. Fast or slow initialization is to fine-tune time, generally not required, just set as 00.
4. Bus BaudRate and bit definition.
5. Auto connection hold in ECU communication, connection message and time can be user-defined.
6. Frame time definition, default by standard setting, time can be changed to accelerate the transport rate.
7. High and low level time in fast initialization may be different for the module other than OBD2. For this specific ECU, K-line logic analysis function of this development toll can be used to obtain the initialization information from the original tester.
8. Bus initialization:

After configuring the above parameters, connect ECU(another demoboard may be used for the analog operation), after starting the bus, select fast or slow initialization for the bus triggering. Note that frame must be 0x81 instruction in fast initialization. Another instruction or null may be along with the slow initialization.

But ISO14230 first formal instruction is generally as 10 xx(StartDiagnosticSession)

9. Select message format in transmission:

0：Single header format, no address information(Mode of operation by a few of manufacturers)

1：9141-2 format, (OBD2-CARB mode of operation)

2：Request by Physical Addr., (Generally used for special diagnosis)

3：Request by Functional Addr.(OBD2 mode of operation)

10. Transmit data other option:

Additional length byte, modify Target Addr. or Source Addr. are optional.

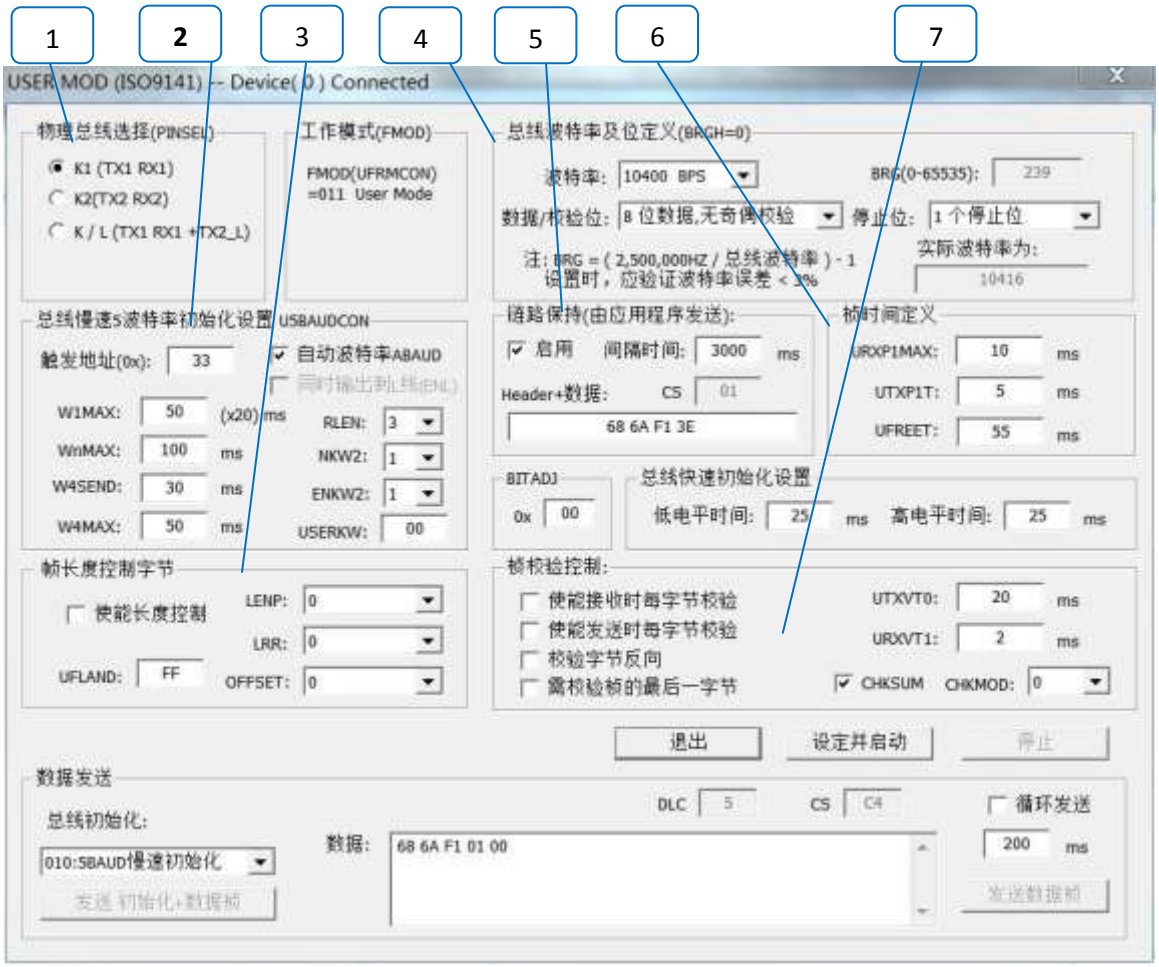
2.11.2 Demoboard Test

Test can be done between two demoboards, one simulates ECU(fast or slow initialization), the other is used for the diagnosis, or directly tested on the vehicle, this ISO14230 is one of the protocols that are extensively used, which is easily found in the vehicle.

2.12 9141-2 Functional Test

ISO9141-2, data frame with no length byte, is different from ISO14230. ISO9141-2 communication is one of FMOD=011 defined modes. Specific setting.

2.12.1 Description of User Interface



1. Select physical bus:
   K1 or K2 can be selected as communication bus, when K1 line is selected, and enable L line used in initialization, only set ON U5BAUDCON ENL bit.
2. 5 Baud setting in initialization, this setting is the same as ISO14230, but the difference is that two buses return different KEYWORDs in initialization.
3. Frame length control setting, ISO9141-2 must not be enabled. The function can be set for the specific application, e.g.: HONDA specific diagnosis protocol, where this interface is used to set the frame length and CheckSum mode for communication purpose.
4. BaudRate and bit definition setting
5. Connection hold setting.

6. Frame time definition
7. Frame check control, each byte check not enabled. CheckSum enable mode

2.12.2 Demoboard Test

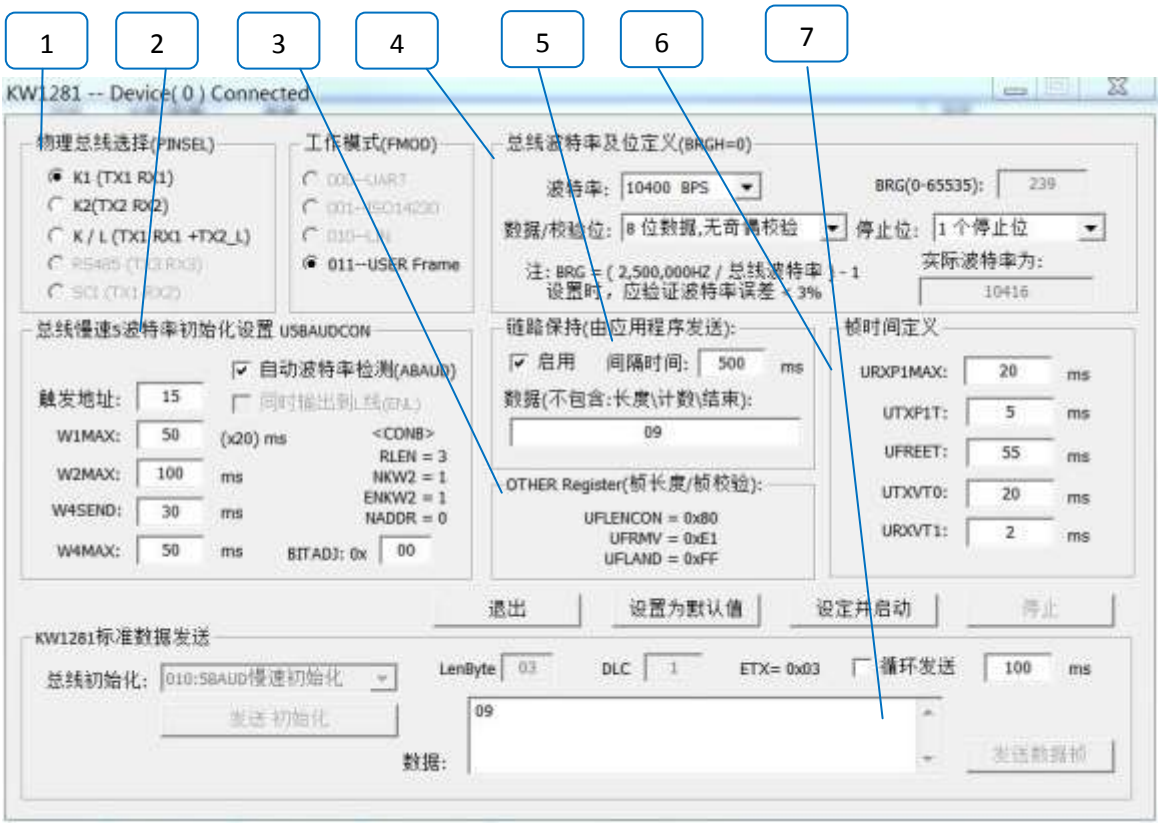
Test can be done between two demoboards, one simulates ECU(ISO9141-2), the other is used for the diagnosis, or two demoboards are directly used to open this window for data exchange.

2.13 KW1281 Functional Test

KW1281 is one of BOSCH diagnosis communication test, which is extensively applied in Volkswagen vehicle.

Different from ISO9141 configuration, KW1281 requires the configuration of frame length, and the reversed check of in-frame each byte.

Like ISO9141-2 communication, it is one specific setting of FMOD=011 defined modes.



1. Select physical bus: The same as ISO9141-2
   K1 or K2 can be selected as communication bus, when K1 line is selected, and enable L line used in initialization, only set ON U5BAUDCON ENL bit.
2. Initialization setting(The only difference with ISO14230-2, NADDR=0)
   In bus initialization, KEYWORD returned from KW1281 is 01 8A.

Trigger address is the address of the module, this trigger address is required to change before communicating with that ECU module, see 8.3.4.1 in Chapter 8 for the trigger address definition of KW1281.

3. Default length control and each byte check control, which are the key to KW1281 protocol setting, see 8.3 in Chapter 8 of ET7190 Data Manual for KW1281 message format and data transmission
4. BaudRate and bit definition setting
5. Connection hold setting
6. Frame time definition, different from ISO9141/ISO14230, the time UTXVT0 and URXVT1 for each byte check is required to set.
7. Transmission of data frame, data just input SID+DAT byte in KW1281 message, ML/MC/ETX is automatically added by this application program.

2.13.1 Demoboard Communication Test

1. Connect K, GND, BATT line to vehicle OBD2 socket, input the address of corresponding module in trigger address, see ET7190 8.3.4.1 in Chapter 8 for the trigger address definition of KW1281.
2. Click “Transmission Initialization”, if ECU is successfully connected, ECU FF about ECU First Frame information can be received, one 09 service is transmitted for the continuous reception of the next frame ECU information until ECU information is completely received.
3. All commands under ET7190 Data Manual 8.3.1 in Chapter 8 can be tested.

2.14 J1708 Functional Test

J1708 is RS-485-based SAE standard, which can be applied in agricultural vehicle, commercial vehicle and heavy machinery, which is released and maintained by SAE(Society of Automobile Engineers). It is primarily used for the serial communication of the ECUs of heavy vehicle or between vehicle and PC.

2.14.1 Description of User Interface

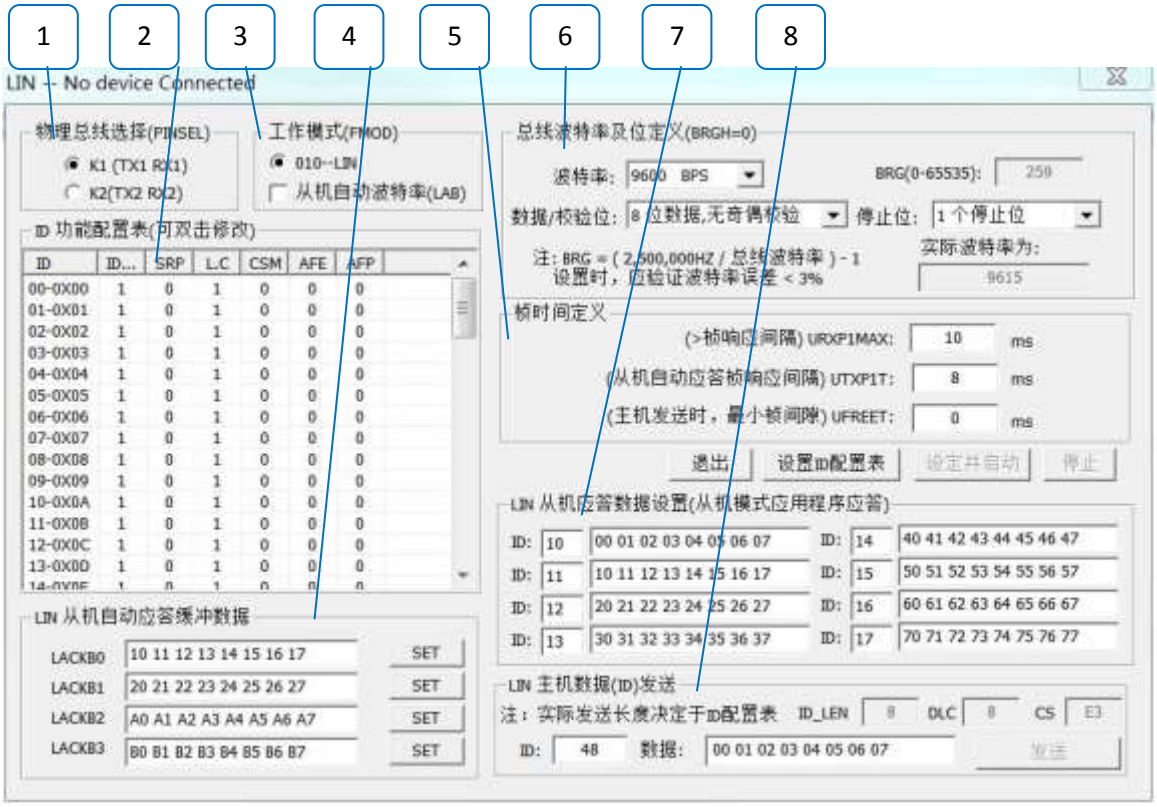
1. Select physical bus
   Theoretically, SAE J1708 physical bus must be RS485 driven bus, but variation application is found domestically: J1708 message format is applied in the physical connection K line, where K line can be also used for J1708 test.
2. Frame control, with no length byte control. CheckSum enable J1708 specific check method.
3. Definition of frame time:
   URXP1T/UTXP1T/UFREET must be set Clear, interbyte space J1708 is always 0, interframe gap is actually controlled by priority control register. Minimum time is 10bits.
4. J1708 BaudRate, must be 9600bps as per this standard

2.14.2 Communication Test

Transmit or receive test can be directly done between two demoboards.

2.15 LIN Functional Test

2.15.1 LIN User Interface



1. Select physical bus: K1 or K2 line can be selected for the operation.
2. Configuration Table of ID function: Define ID function, ID identifier functional definition of the used node is the key to the normal operation of one LIN network, when ID identifier function is defined, any node ID cannot cause conflict. ET7190 defines ID function by setting LIDCON control register set. Double click List Item to modify ID Configuration Table:

3. Mode of operation: FMOD=010 select LIN protocol UFRMCON LAB as slaver, which can be set ON, by which each byte receives auto BaudRate. But as ET7190 application, LAB is generally cleared, because ET7190 uses the crystal oscillator with highly precise clock. Auto BaudRate is not required.
4. The value setting of 4 LIN slaver auto respond buffers, this program only writes when the setting is started, which doesn't dynamically change the function, and user can change the value of buffer at any time in actual operation.
5. The definition of these three times determines the maximum time of LIN frame slot. Required to ensure: URXP1T>UTXP1T.
6. Bus BaudRate setting. Lin generally uses one of 9600, 10400 and 19200bps. When enable auto BaudRate is used, actual UBRG value can change the definition of frame time.
7. For respond test as slaver, this data is transmitted by the application program, actual transmit length is defined by ID.LENM set value in Configuration Table.
8. Master: Transmit master frame or start slaver task(When slaver frame task is started, data is empty)

   Slaver: Master task must not be transmitted, but only respond to slaver task. Or receive master task, slaver task.

2.15.2 LIN Send Respond Experiment

e.g.: Node A used as master node, start slaver task ID=10, 11, transmit master task ID=12. Because we only use 3 IDs, so these 3 IDs are set respectively. (All other IDENs can be set Clear, because no other node exists in the network, neither any other ID will be transmitted, so no setting affects nothing)

Node B is used as slaver node, these 3 IDs are set respectively in the same way.

1. **ID Setting of Master Node A:**

09-0X09	1	0	1	0	0	0
10-0X0A	1	1	3	0	1	3
11-0X0B	1	1	3	0	0	0
12-0X0C	1	0	3	0	0	0
13-0X0D	1	0	1	0	0	0

Notes:

- ID: For 10/11/12 IDEN=1, which indicates that these 3 IDs are valid.
  ID：For 10/11 SRP=1, which indicates that slaver respond is required, and this is slaver task.
  ID: For 12 SRP=0, which indicates master task, all data are sent from master and received by slaver.
- ID: For 10/11/12, LENM=3, which indicates that the frame length of 3 IDs is 8 bytes.
- ID: For 10/11/12 CSM=0, which indicates common CheckSum mode LIN1 supported ID.
- ID: Because this is master, the followed AFE AFP is respond data, which is actually irrelevant setting.

2. ID Setting of Slaver Node B:

09-0X09	1	0	1	0	0	0
10-0X0A	1	1	3	0	1	3
11-0X0B	1	1	3	0	0	0
12-0X0C	1	0	3	0	0	0
13-0X0D	1	0	1	0	0	0

Notes:

- ID: For 10/11/12 IDEN=1, which indicates that these 3 IDs are valid.
  For 10/11 SRP=1, which indicates that slaver respond is required, and this is slaver task.
  ID: 12 SRP=0, which indicates master task, all data are sent from master, and this slaver only receives but not responds.
- ID: For 10/11/12, LENM=3, which indicates that the frame length of 3 IDs is 8 bytes, with the same length value as master.
- ID: For 10/11/12 CSM=0, which indicates common CheckSum mode LIN1 supported ID, the same with master.
- ID: 10 Because this slaver respond is required, respond select auto AFE=1, respond data is LACKB3(AFP=3).
  ID: 11 Because this slaver respond is required, respond select user transmit AFE=0, AFP value is irrelevant.
  ID: 12 Because this slaver respond is not required, AFE, AFP set value are irrelevant, only receives.

If carefully examined, Node A and Node B are identical in setting, but each ID processed at

multiple slaver nodes is different, so the setting may be varied. And AFE, AFP respond setting at Master Node A are invalid. If master node is required to transmit data, master task is directly used for the transmission with no need of response.

3. Send Test:

- Master starts slaver task 10(Input 10 in ID Click Send): In master interface, it can be seen that master receives one frame data, data value is the value of LACKB3 Buffer at Slaver Node B. Slaver interface shows no information, because this data is automatically transmitted by ET7190 from the buffer.
- Master starts Slaver Task 11(Input 11 in ID Click Send): Likewise in master interface, it can be seen that master receives one frame data, data value is the value of slaver respond data area ID=11 at Slaver Node B(User respond value). One frame data is sent in the user interface.

- Master sends Master Task 12(Input 12 in ID): At that time, input the intended data with the mandatory length in 8 bytes(Comply with definition in ID Control Table). Click Send. Slaver will receive the data frame.

- ET7190 actually doesn't differentiate master from slaver in hardware, the definition of master or slaver is entirely user-defined, in this experiment, Node B is identical to Node A, where you can launch the frame header. Send master task. But only one node is used as master node to start the slaver task or send master task in software when user plans the network.

2.16 UART Functional Test

Only one simple internal test procedure, when FMOD=000 is tested, the universal UART is used for receive or send test. This is not described for there is no real application.

Chapter 3 RS232COMM Development Example VC++6.0 Code

In software kit, RS232COMM content is a VC++ 6.0 open source code, which is based on PC RS232 serial communication port, where ISO9141-2/ISO14230-SLOW/ISO14230-FAST/J1850VPW /J1850 VPW/ISO15765-11bits/ISO15765-29bits/KW1281 protocol is demonstrated for the diagnosis of C++ source code.

3.1 RS232COMM Relevant Functions

3.1.1 Worker Threads in Communication

- 3 worker threads in the given example correspond to ET7190 EUART/J1850/CAN module respectively:
  UINT Thread_EUART(LPVOID pParam);
  UINT Thread_J1850(LPVOID pParam);
  UINT Thread_CAN(LPVOID pParam);
- The effect of worker thread is to receive or transmit message via PC RS232.
- In worker thread, this application has not added the error handling. When error flag is received, user may process the content by the required additional error.
- When message frame is received, this thread only adds the received message with no processing to ItemList for display.

3.1.2 Function Arrangement in Communication Class

There are 4 functions in each communication class:
SetMainDlgCtrl(): Dialog control when this protocol is set
SetET7190Register() : ET7190 register when this protocol is set, user should thoroughly understand this configuration function.

StartCommunication(): Start ET7190 module and the corresponding communication thread.
StopCommunication(): Stop ET7190 module and the corresponding communication thread.

Note: ISO9141-2/ISO14230-SLOW/ISO14230-FAST/KW1281 uses the same communication worker thread: Thread_EUART. And these protocols apply ET7190 EUART module, and the differential setting should be carefully understood.

3.2 About OBD2 Diagnosis

Auto search function of OBD2 diagnosis is implemented by understanding the diagnostic setting of each bus, testing 01 00 requests of each bus one by one, when the correct response is tested, current protocol is used for other request service.

3.2.1 Mandatory ID Requirement for OBD2 Diagnosis

Bus	Tester ID			ECU Respond ID			Data Length	Remark
	type	tgt	src	type	tgt	src		
ISO9141/J1850 10.4K	68	6A	F1	48	6B	Ecu addr	7	9141 Init Addr:0x33
ISO14230 10.4kbps	11LL LLLLb	33	F1	10LL LLLLb	F1	Ecu addr	7	Slow or Fast Init
J1850 PWM 41.6kbps	61	6A	F1	41	6B	Ecu addr	7	
CAN 29bits	18 DB	33	F1	18 DA	F1	Ecu addr	8	250KB/500KB
CAN 11bits	7DF			7E8~7EF			8	250KB/500KB

3.2.2 Notes for OBD2 Diagnosis

1. OBD2 is Standard ISO15031-5 of the application layer, which gives the unified definition to diagnosis data.

2. The bottom layer of communication protocol includes, J1850VPW/PWM /ISO14230 5buad / ISO14230 Fast/ISO9141-2 / ISO15765 11bits/29bits, all search processes may require 9 cases of test protocol to the maximum. (In CAN, 250KBPS and 500KPBS is tested respectively, ISO14230 requires two cases: fast and slow initialization)

3. In order to search the protocol as soon as possible, it is recommended to search the protocol in sequence:
   ISO15765_11bits_500kbps → ISO15765_29bits_500kbps → ISO14230_FAST→ J1850_VPW → J1850PWM → ISO9141_2 → ISO14230_SLOW → ISO15765_11bits_250kbps → ISO15765_29bits_250kbps
   Note: One 5 Baud slow initialization takes 3 seconds at least. If OBD2 application cannot be searched, the entire process requires about 10S. 250KBPS CAN is rarely applied, so it is put in the last sequence of search.

4. When OBD2 auto find test is carried out in the vehicle of unknown communication protocol, Note: In ISO15765 test, Listen Only Mode must be firstly used to carry out the bus BaudRate test.
   Because CAN bus permits no different BaudRate configurations in the same bus, or

CAN transreceiver will continuously send error frame, and any node in the bus is not allowed for communication. This will result in ECU storage bus TroubleCode. Listen Only Mode will not send error frame. In event of incorrect BaudRate, CAN error flag will be received.

5. Likewise, J1850 VPW/PWM uses PIN2 at the same OBD2 interface, vehicle bus always has data in communication. If in VPW(PWM) bus, you begin to transmit PWM(VPW) data, vehicle ECU will receive the error, so the bus must be listened before testing to confirm the bus type.
   When J1850 listens, all YZZCONs are set Clear, and IFR respond is off. All HDFILTER[]s are stuffed as 0x07, and all data are received. If the bus is correct, ET7190 can receive bus data frame, but if the bus is incorrect, JERRF error flag bit will be set.

6. In CAN and J1850 bus test, if there is no data in the bus, it will not be determined in testing. Normally, J1850 must have some data in transport in the vehicle. If CAN passes gateway diagnosis, no data may exist in normal bus, but in event of no data, it can be deemed as correct BaudRate for testing.