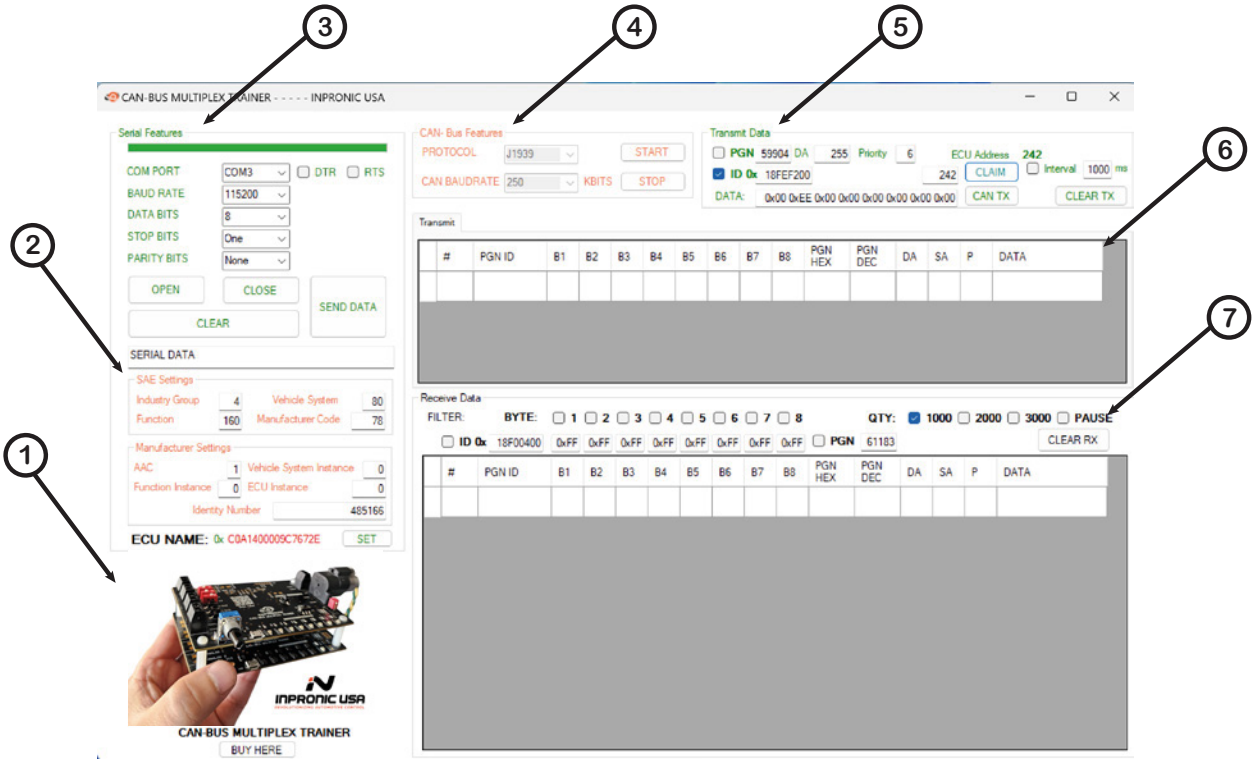
Can-Bus Multiplex Trainer SOFTWARE

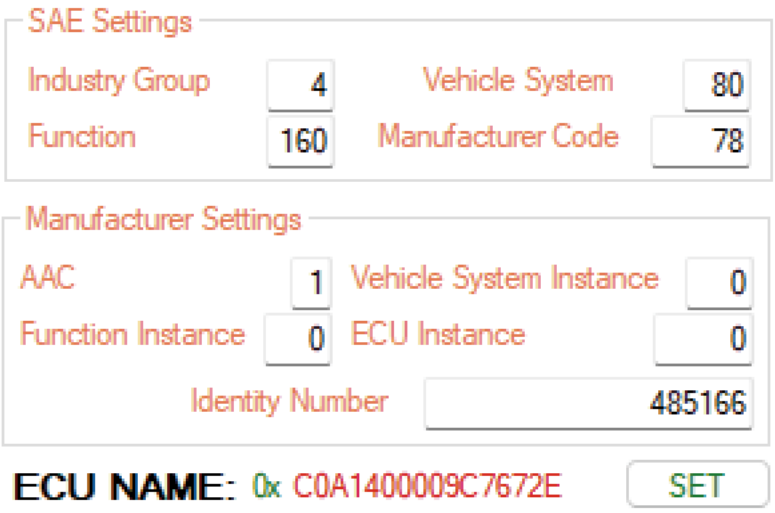


1. Can bus Multiplex Trainer Board.

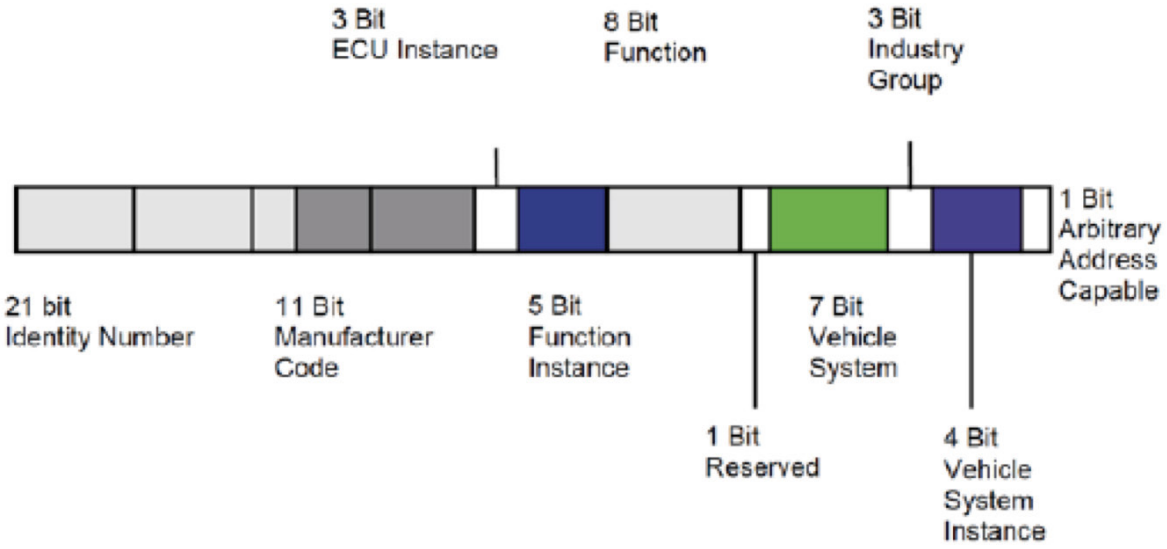
If you don't have already the board you can buy here:
https://inpronicusa.com/store/ols/products/j1939-can-bus-multiplex-trainer-kit

4101 Cox Rd, Suite 200-12 Glen Allen VA, 23060
sales@inpronicusa.com

2. SAE Settings J1939

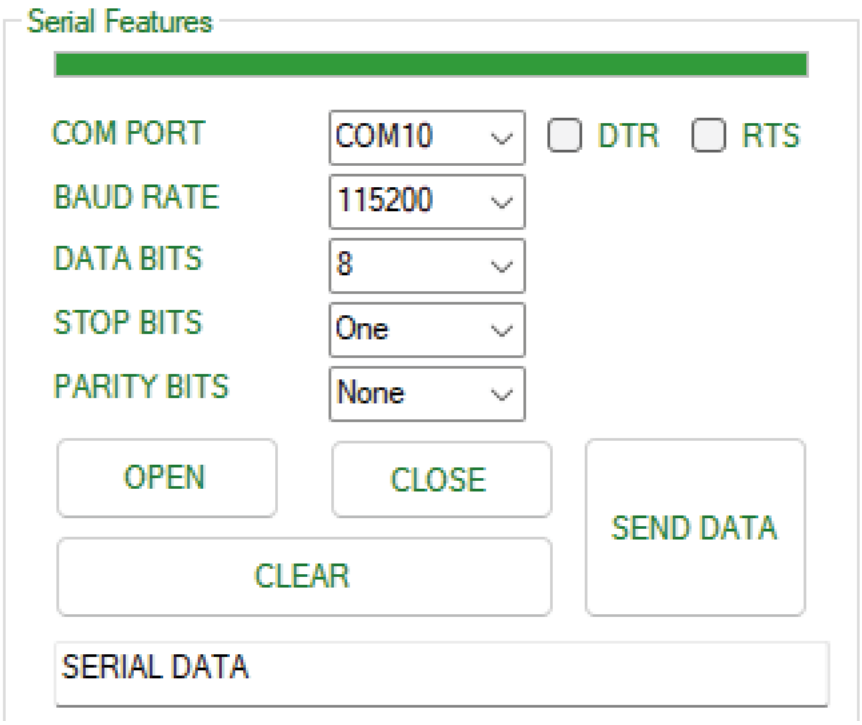


Each ECU in a J1939 protocol must hold at least one NAME and one address, SAE J1939/81 defines a 64 bit for NAME, showing bellow.



Add all information accordingly and push SET to obtain the information to CLAIM the address for J1939 Protocol.

3. Serial Features



In this Tab you can configure the serial protocol for communicate with the CAN-BUS Multiplex Trainer board under USB.

First choose the COM port connected with the board in this case it is COM10, set the BAUD RATE to 115200, data bits 8, stop bit one, and choose no parity.

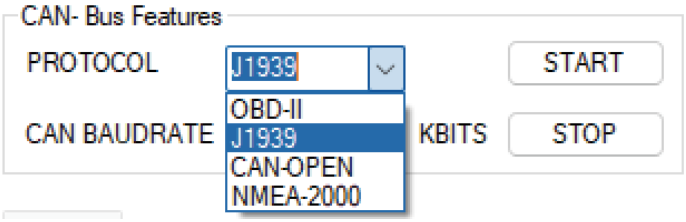
Select OPEN to start the serial communication with the CAN-Bus Multiplex Trainer.

Don't forget to STOP the CAN-BUS before CLOSE The Serial Port, if you need send a serial data to the board you can do that using the Text box SERIAL DATA you can CLEAR it and write any text and SEND DATA.

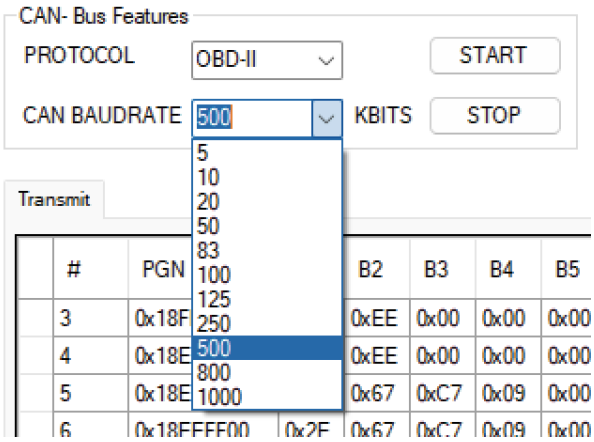
It is very useful to deploy a new products special when you need to connect other peripherals like HMI screens, LCD, and so on.

4. CAN-Bus Features

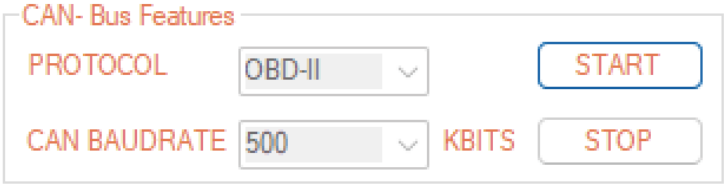
In this Tab you can choose the Protocol to Monitoring – Simulate or Process the Data.



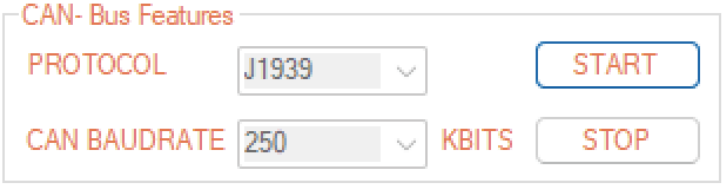
Also you can choose the Baud rate of CAN, typically the OBD-II use a 500Kbps, and J1939 250Kbps, and so on.!



After choose this parameters you can START the communication under CAN-BUS Protocol



OBD-II PROTOCOL



J1930 PROTOCOL

5. Transmit Data

In this Tab you can choose the IDs or PIDs accordingly your protocol.

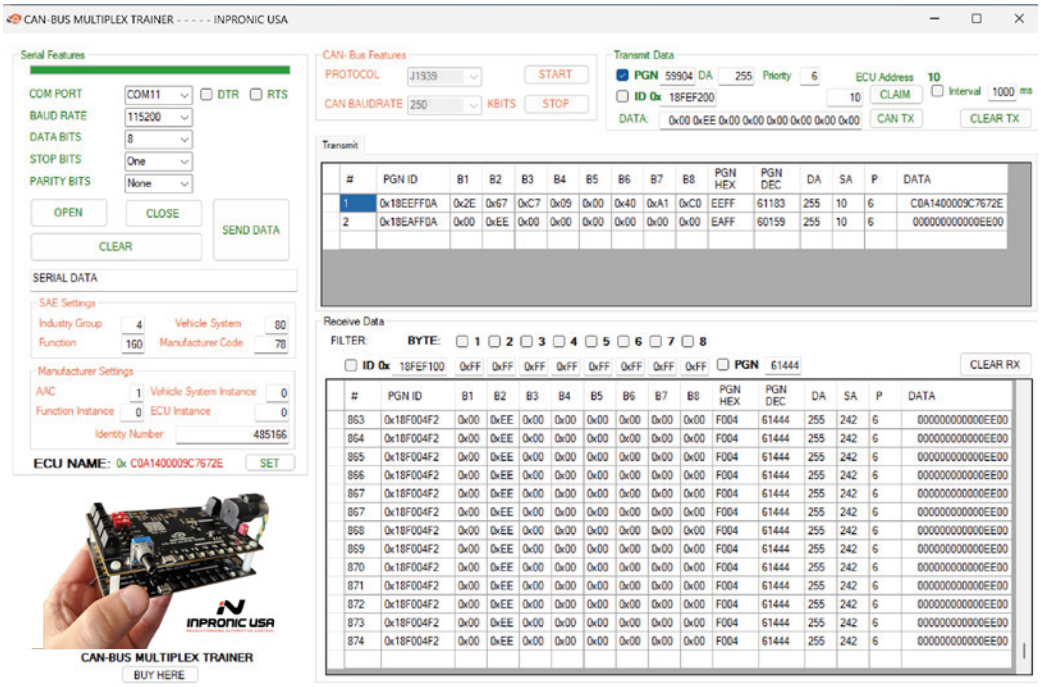
For example, for J1939 you have three different ways to send data.

First one is under PGNs, SA, DA, P, but to do it, is important to CLAIM the address to the ECU, in this case you first need to SET the name of the ECU and the address, if you have already other ECU with the same Address our TRAINER will change the actual ECU Address accordingly the J1939 Protocol.

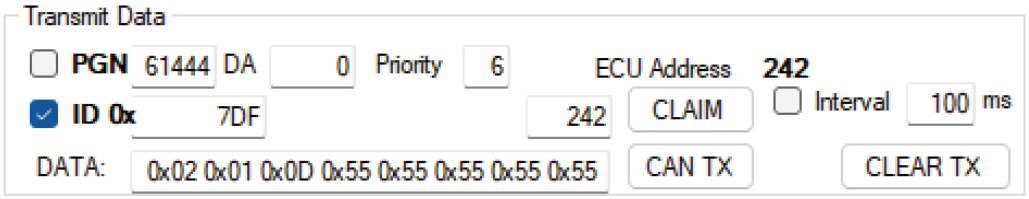
Second one, if you know the PGN ID you can choose it and of course select the DATA to need to be transmitted. And press CAN-TX.

Third you can select if need to send under PGNs or under ID, and select the value of interval from 1ms to 65559ms and the data will send accordingly your selection.

When you send the information on the Trainer Board you must receive the information on the other side like this.



Under OBD-II you just must send the PGN ID and the Data, for example I'll request a speed for the vehicle for this purpose I need to know the PID information, in this case it is 0x0D, and need to request, the request address under this protocol is 7DF, and the Data is 0x02 0x01 0x0D 0x55 0x55 0x55 0x55 0x55.



When you send this information from OBD-II to the vehicle it will response with the speed information under PGN ID 7E8 with data 0x03 0x41 0x04 0x0F 0xAA 0xAA 0xAA 0xAA. It corresponds to a 4Km/h

#	PGN ID	B1	B2	B3	B4	B5	B6	B7	B8	PGN HEX	PGN DEC	DA	SA	P	DATA
11	0x000007E8	0x03	0x41	0x0D	0x13	0xAA	0xAA	0xAA	0xAA	7	7	7	232	0	AAAAAAAA130D4103
12	0x000007E8	0x03	0x41	0x0D	0x15	0xAA	0xAA	0xAA	0xAA	7	7	7	232	0	AAAAAAAA150D4103
13	0x000007E8	0x03	0x41	0x0D	0x16	0xAA	0xAA	0xAA	0xAA	7	7	7	232	0	AAAAAAAA160D4103
14	0x000007E8	0x03	0x41	0x0D	0x14	0xAA	0xAA	0xAA	0xAA	7	7	7	232	0	AAAAAAAA140D4103
15	0x000007E8	0x03	0x41	0x0D	0x04	0xAA	0xAA	0xAA	0xAA	7	7	7	232	0	AAAAAAAA040D4103

6. Transmit Tab

In this Tab, you will see all of the data sending by CAN-BUS, under the different protocol explained before. In this example bellow you can see the request data under OBD-II for speed and rpm for vehicle.

Transmit

#	PGN ID	B1	B2	B3	B4	B5	B6	B7	B8	PGN HEX	PGN DEC	DA	SA	P	DATA
2	0x000007DF	0x02	0x01	0x0D	0x55	0x55	0x55	0x55	0x55	7	7	7	223	0	55555555550D0102
3	0x000007DF	0x02	0x01	0x0D	0x55	0x55	0x55	0x55	0x55	7	7	7	223	0	55555555550D0102
4	0x000007DF	0x02	0x01	0x0C	0x55	0x55	0x55	0x55	0x55	7	7	7	223	0	55555555550C0102
5	0x000007DF	0x02	0x01	0x0C	0x55	0x55	0x55	0x55	0x55	7	7	7	223	0	55555555550C0102
6	0x000007DF	0x02	0x01	0x0C	0x55	0x55	0x55	0x55	0x55	7	7	7	223	0	55555555550C0102

7. Receive and Filter Tab

In this box you can see all the data received under CAN-Bus and under you protocol selected. You can filter the information for PGN, PGN ID, Data bytes. It is useful when you want to see for example just the response from OBD-II, because If you don't filter it is a lot of messages coming faster, so you need to filtering to just check the parameters what you are looking for.

Receive Data

FILTER: **BYTE:** ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8

☑ **ID 0x** 7E8 | 0xFF | 0xFF | 0xFF | 0xFF | 0xFF | 0xFF | 0xFF | 0xFF ☐ **PGN** 61444 CLEAR RX

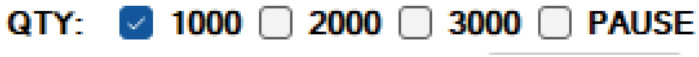
#	PGN ID	B1	B2	B3	B4	B5	B6	B7	B8	PGN HEX	PGN DEC	DA	SA	P	DATA
11	0x000007E8	0x03	0x41	0x0D	0x13	0xAA	0xAA	0xAA	0xAA	7	7	7	232	0	AAAAAAAA130D4103
12	0x000007E8	0x03	0x41	0x0D	0x15	0xAA	0xAA	0xAA	0xAA	7	7	7	232	0	AAAAAAAA150D4103
13	0x000007E8	0x03	0x41	0x0D	0x16	0xAA	0xAA	0xAA	0xAA	7	7	7	232	0	AAAAAAAA160D4103
14	0x000007E8	0x03	0x41	0x0D	0x14	0xAA	0xAA	0xAA	0xAA	7	7	7	232	0	AAAAAAAA140D4103
15	0x000007E8	0x03	0x41	0x0D	0x04	0xAA	0xAA	0xAA	0xAA	7	7	7	232	0	AAAAAAAA040D4103
16	0x000007E8	0x03	0x41	0x0D	0x05	0xAA	0xAA	0xAA	0xAA	7	7	7	232	0	AAAAAAAA050D4103
17	0x000007E8	0x03	0x41	0x0D	0x09	0xAA	0xAA	0xAA	0xAA	7	7	7	232	0	AAAAAAAA090D4103
18	0x000007E8	0x03	0x41	0x0D	0x10	0xAA	0xAA	0xAA	0xAA	7	7	7	232	0	AAAAAAAA100D4103
19	0x000007E8	0x03	0x41	0x0D	0x15	0xAA	0xAA	0xAA	0xAA	7	7	7	232	0	AAAAAAAA150D4103

Easy to check and understand.!

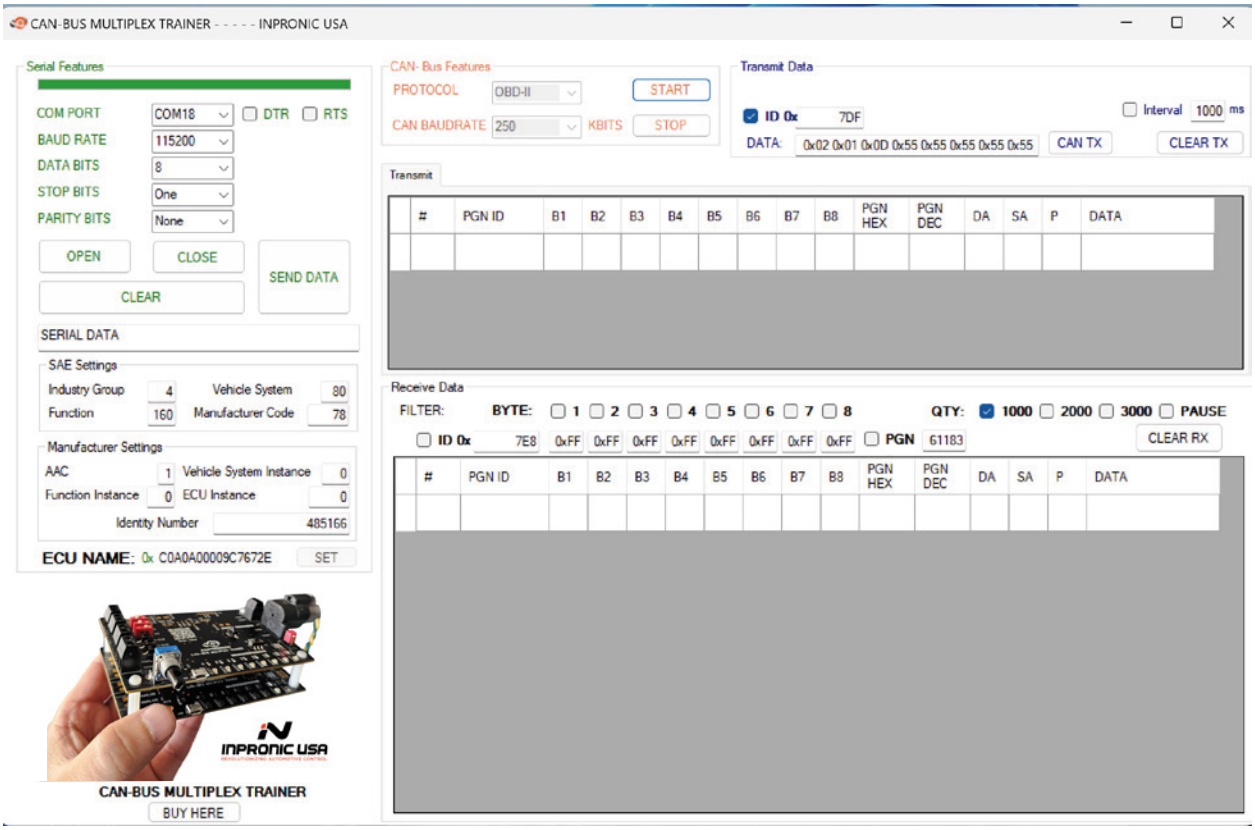
You can use two different windows with the same app, to simulate, it mean the up board can be your reader, and the down board can be your transmitted, and you will be ready to understand how it work before to connect the CAN-BUS Multiplex Trainer to the real world.!

WHAT ARE THE NEW CHANGES IN THIS VERSION V1.0.2.

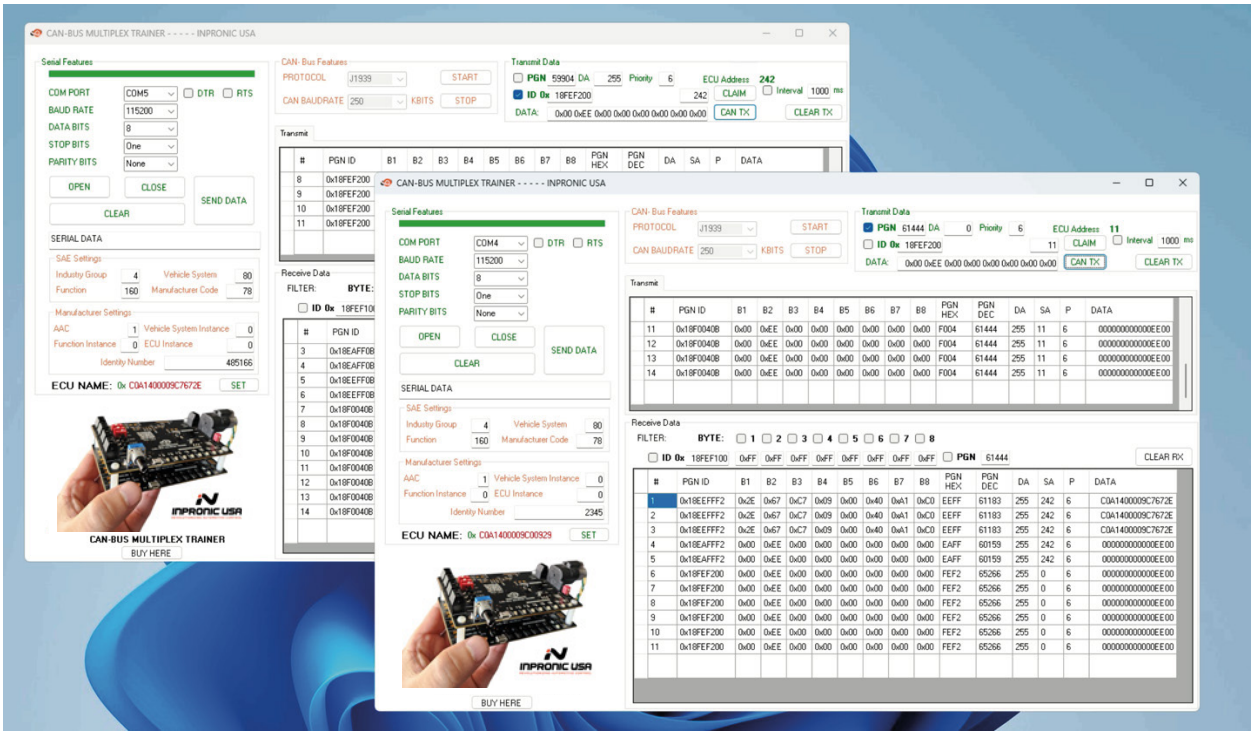
1. We have implemented three configurable Data Frame sizes—1000, 2000, and 3000 rows—to efficiently handle the high volume of data transmitted every millisecond when the board is connected to OBD2, J1939, or NMEA2000 networks. This approach ensures that the software remains responsive and avoids performance bottlenecks. Additionally, we've included a PAUSE function, allowing users to temporarily halt data reception for detailed analysis without interrupting ongoing CAN-Bus communication.



2. We have optimized the data acquisition process from USB to the CAN-Bus software to prevent crashes that occasionally occurred due to high transmission speeds. These improvements were implemented solely within the CAN-Bus Trainer software, eliminating the need for any firmware updates.



INPRONIC USA
REVOLUTIONIZING AUTOMOTIVE CONTROL



For more information about our Trainer and excited news keep in touch with us.

Any additional information Contact us:

sales@inpronicusa.com

8049281824

4101 Cox Rd, Suite 200-12 Glen Allen VA, 23060
sales@inpronicusa.com