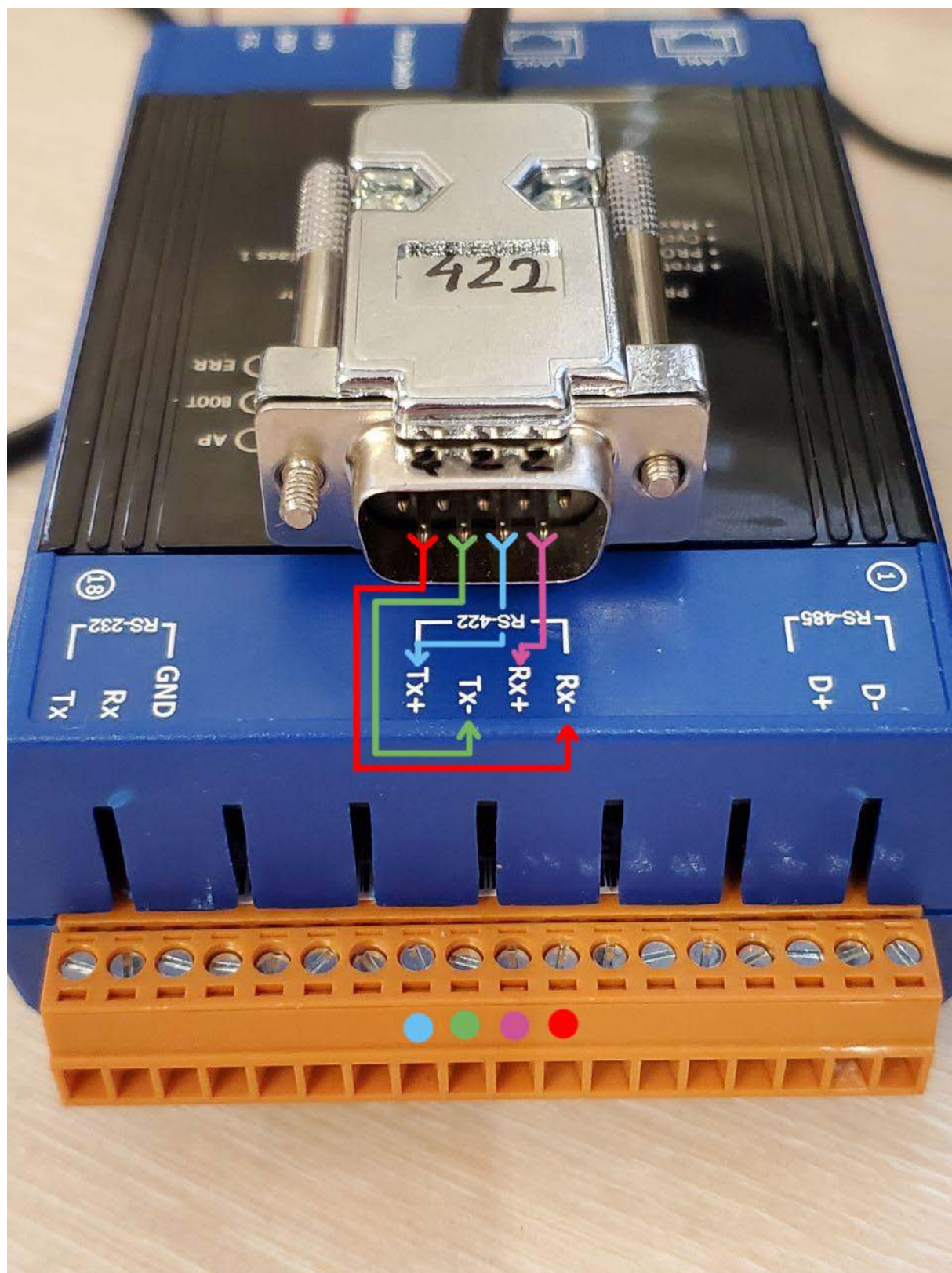
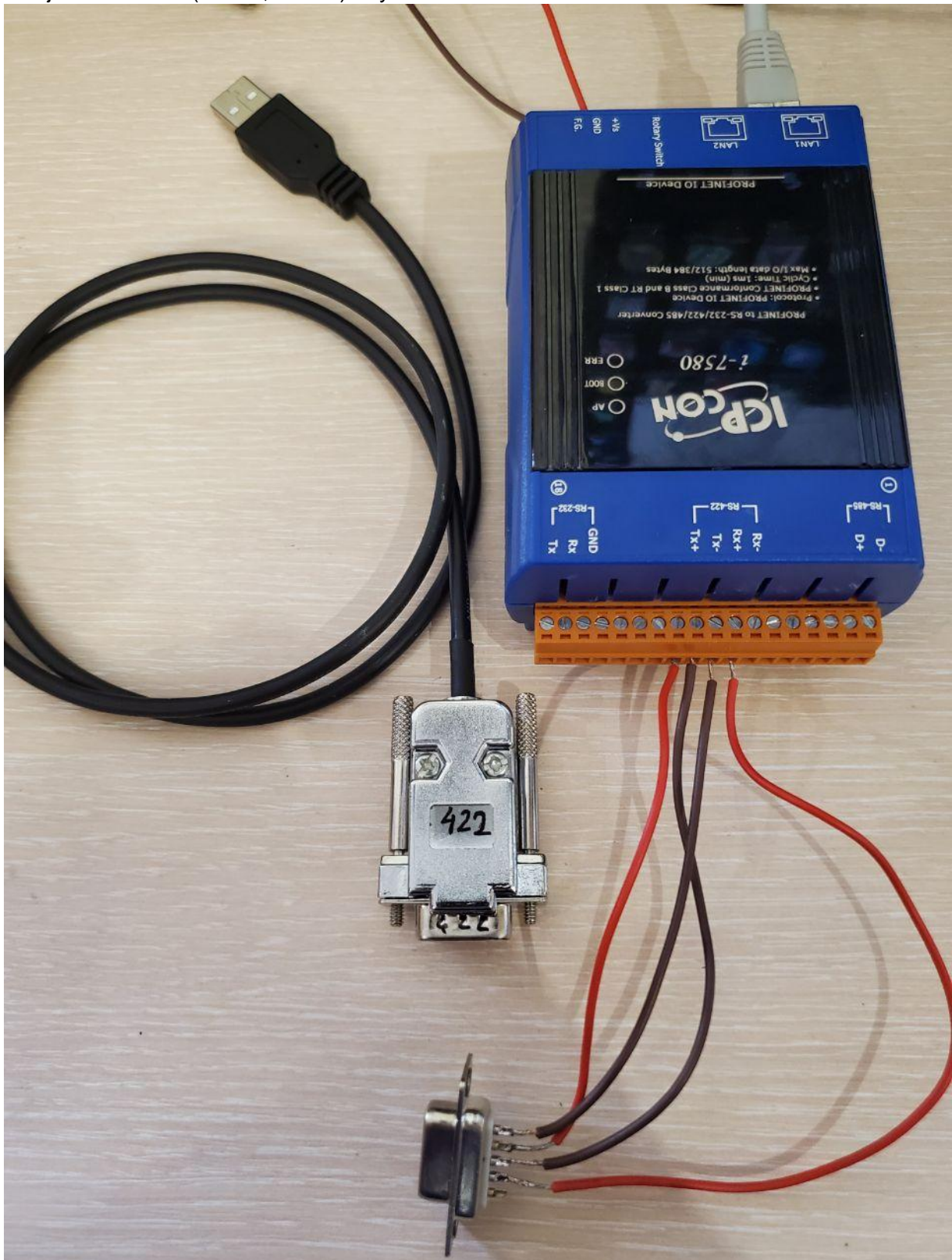


Little help in connecting the converter and checking its performance

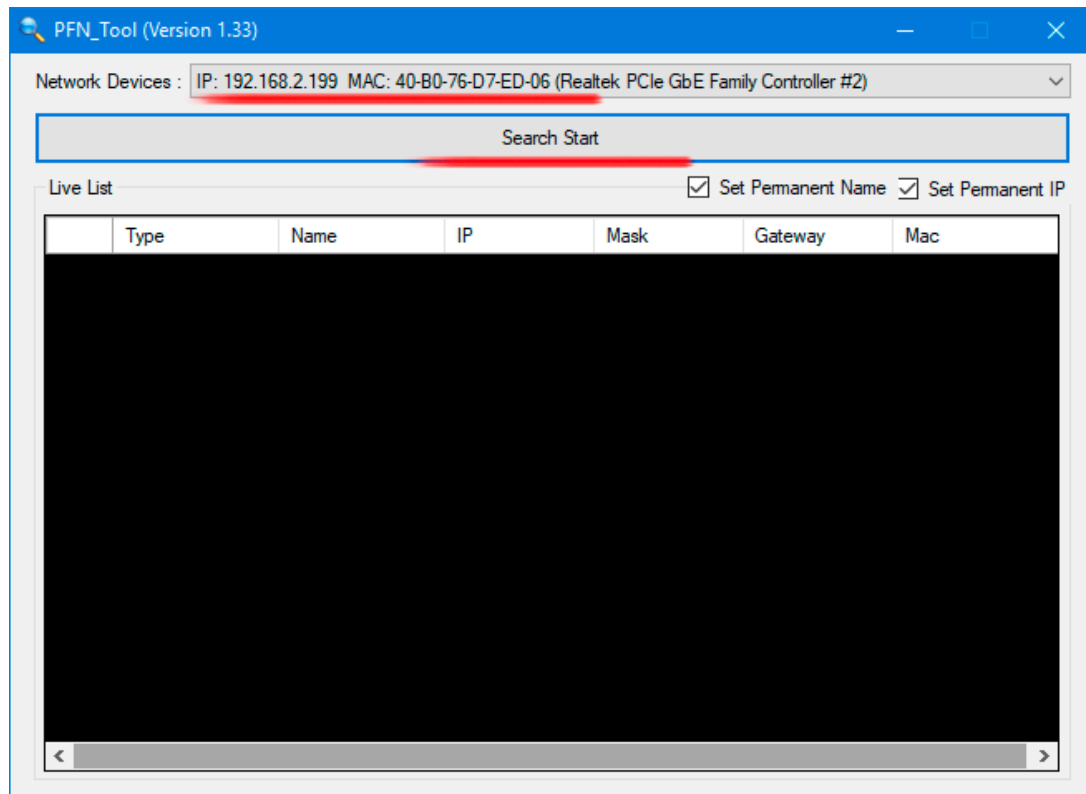


How to test the device

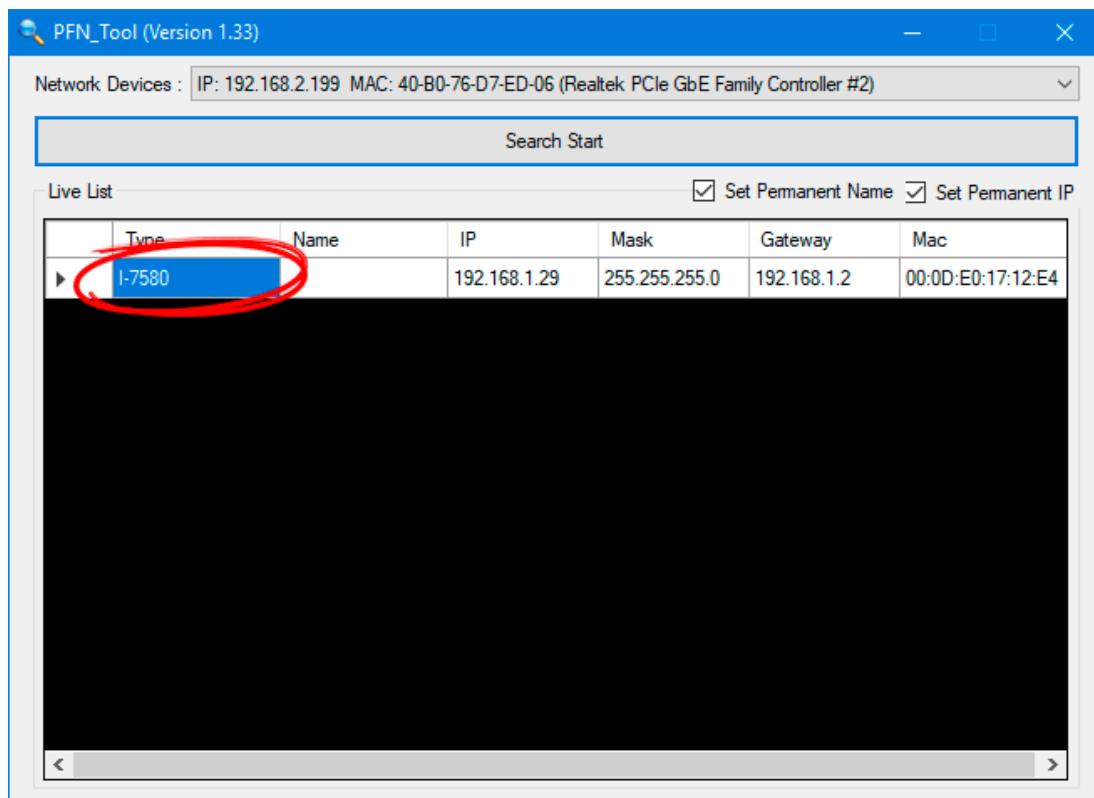
1) First, you should to apply 24V power to the I-7580 converter and then to connect converter via rj45 and rs422 (or 232, or 485) to your PC.



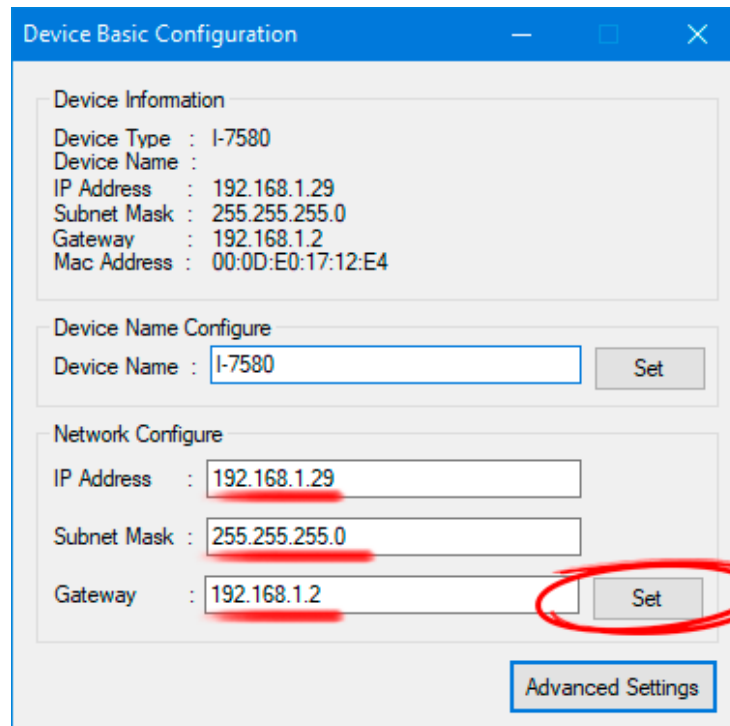
- 2) Open PFN_Tool software and start searching as shown in the pictures



- 3) Select the discovered device by double clicking



4) Configure the device according to your adapter settings



The 'Device Basic Configuration' dialog box is shown. It has three main sections: 'Device Information', 'Device Name Configure', and 'Network Configure'. The 'Device Information' section lists: Device Type: I-7580, Device Name: (empty), IP Address: 192.168.1.29, Subnet Mask: 255.255.255.0, Gateway: 192.168.1.2, and Mac Address: 00:0D:E0:17:12:E4. The 'Device Name Configure' section has a text box for 'Device Name' containing 'I-7580' and a 'Set' button. The 'Network Configure' section has text boxes for 'IP Address' (192.168.1.29), 'Subnet Mask' (255.255.255.0), and 'Gateway' (192.168.1.2), each with a corresponding 'Set' button. A red circle highlights the 'Set' button for the Gateway. At the bottom right is an 'Advanced Settings' button.

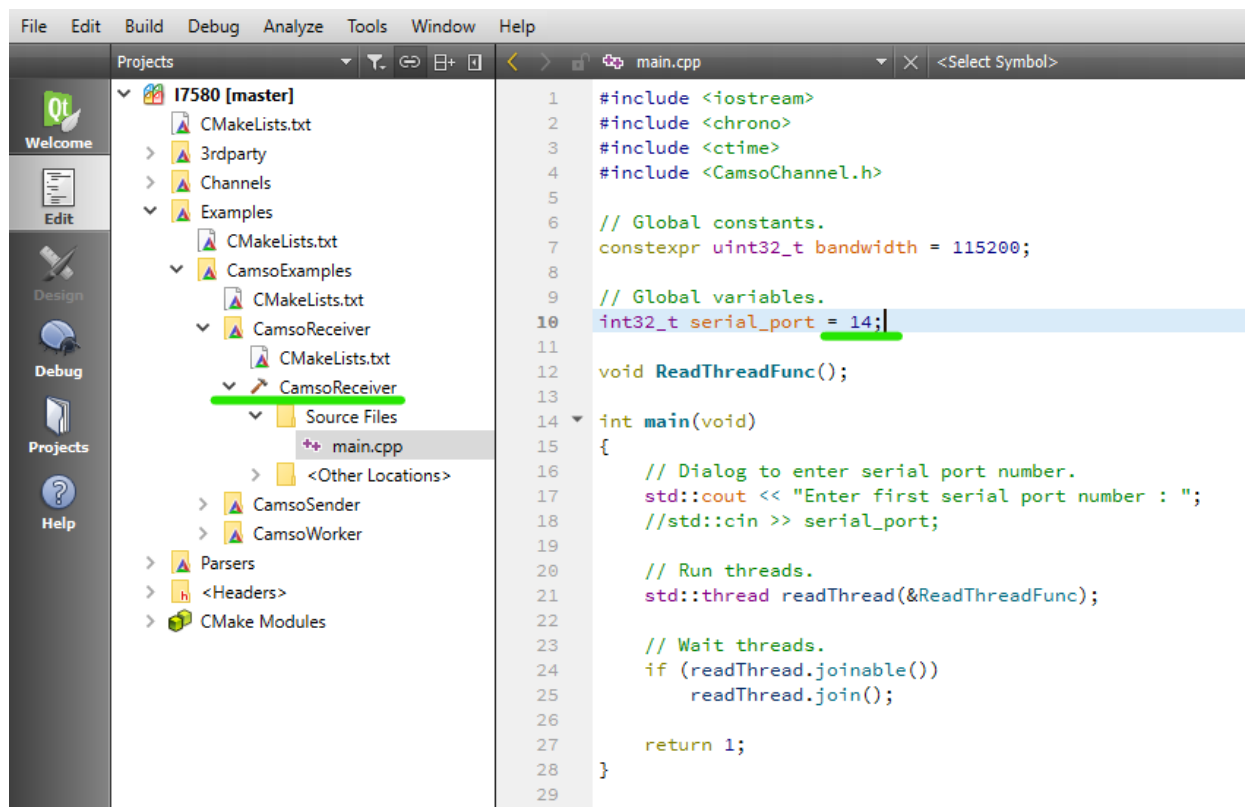
Device Information	
Device Type	I-7580
Device Name	
IP Address	192.168.1.29
Subnet Mask	255.255.255.0
Gateway	192.168.1.2
Mac Address	00:0D:E0:17:12:E4

Device Name Configure	
Device Name	I-7580

Network Configure	
IP Address	192.168.1.29
Subnet Mask	255.255.255.0
Gateway	192.168.1.2

Advanced Settings

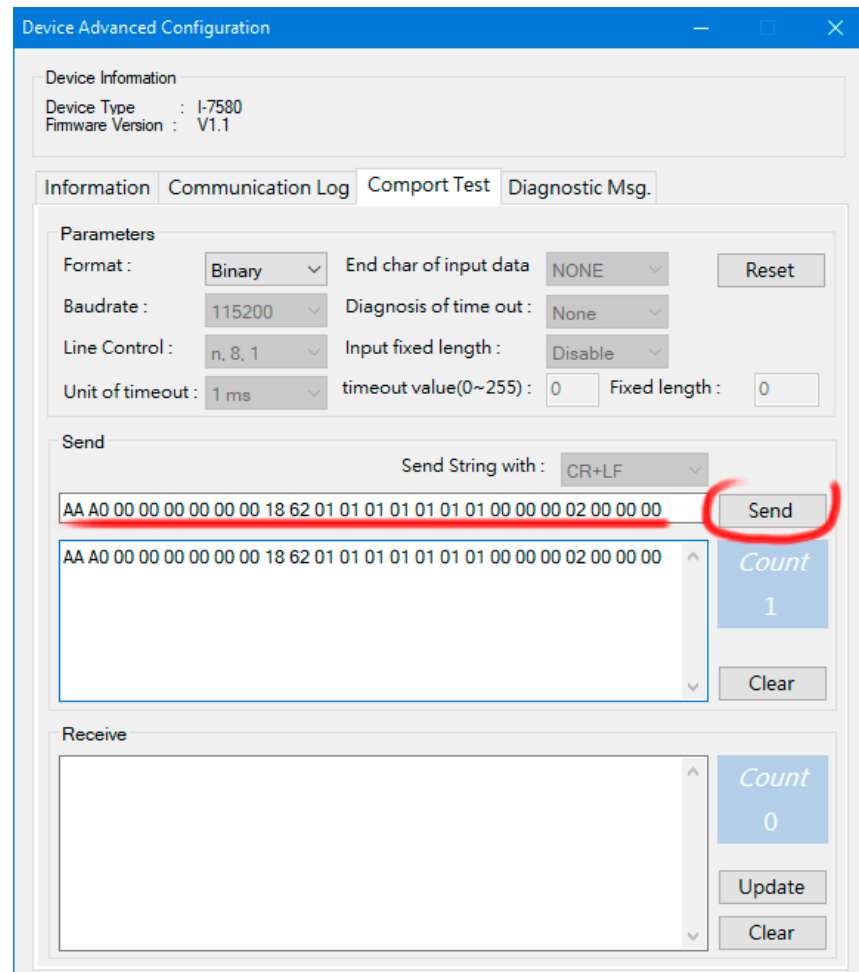
5) Run the sample receiver program (before that, specify in the program the number of the com-port to which the converter is connected)



The screenshot shows the Qt IDE interface. The left sidebar displays the project structure for 'I7580 [master]'. The 'CamsoReceiver' project is selected, showing its source files: 'main.cpp'. The main editor window displays the code for 'main.cpp'. The code includes headers for `<iostream>`, `<chrono>`, `<ctime>`, and `<CamsoChannel.h>`. It defines a global constant `bandwidth = 115200` and a global variable `serial_port = 14`. The `ReadThreadFunc()` function is defined, and the `main()` function calls `ReadThreadFunc()` and waits for the thread to finish. The `serial_port` variable is highlighted in blue.

```
1 #include <iostream>
2 #include <chrono>
3 #include <ctime>
4 #include <CamsoChannel.h>
5
6 // Global constants.
7 constexpr uint32_t bandwidth = 115200;
8
9 // Global variables.
10 int32_t serial_port = 14;
11
12 void ReadThreadFunc();
13
14 int main(void)
15 {
16     // Dialog to enter serial port number.
17     std::cout << "Enter first serial port number : ";
18     //std::cin >> serial_port;
19
20     // Run threads.
21     std::thread readThread(&ReadThreadFunc);
22
23     // Wait threads.
24     if (readThread.joinable())
25         readThread.join();
26
27     return 1;
28 }
29
```

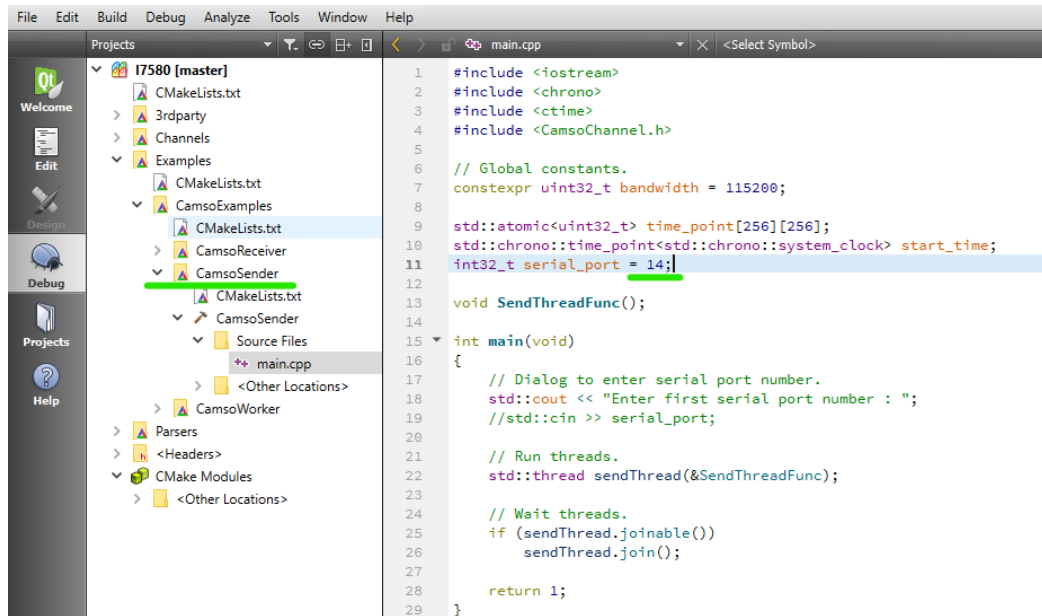
6) If you send this data (AA A0 00 00 00 00 00 00 18 62 01 01 01 01 01 01 00 00 00 02 00 00 00) from I-7580 Converter to PC:



Then in the sample receiver program you can see this output information

```
Application Output
ProfinetReceiver
13:39:00: Debugging starts
Enter first serial port number : 14
Start reading...
No input data
No input data
No input data
port 14 14 bytes
Calibration Start
Zero Positioning Start
Measuring Start
Abort Measure
Fault Reset
New Data Transfer (measuring_start_position = 1, calibration_diameter = 2)
```

7) You can also check the sending of data in the opposite direction, from the PC to the converter. For this you can run the sample sender program (before that, specify in the program the number of the com-port to which the converter is connected)

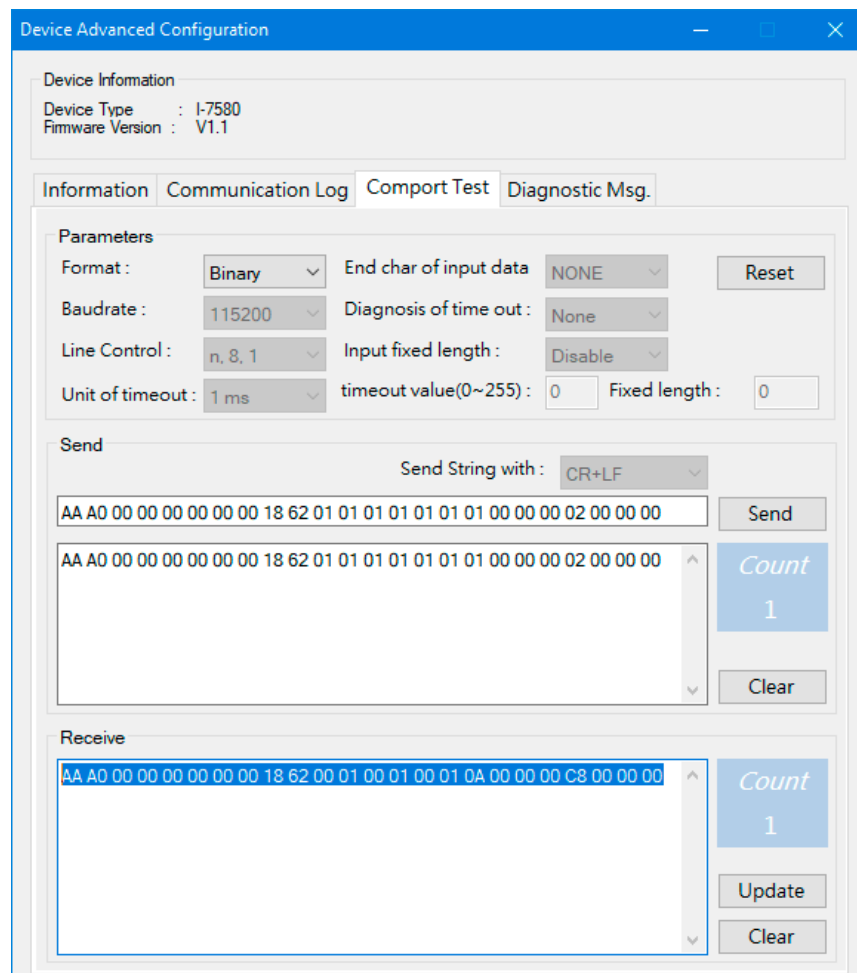


```

1  #include <iostream>
2  #include <chrono>
3  #include <ctime>
4  #include <CamsoChannel.h>
5
6  // Global constants.
7  constexpr uint32_t bandwidth = 115200;
8
9  std::atomic<uint32_t> time_point[256][256];
10 std::chrono::time_point<std::chrono::system_clock> start_time;
11 int32_t serial_port = 14;
12
13 void SendThreadFunc();
14
15 int main(void)
16 {
17     // Dialog to enter serial port number.
18     std::cout << "Enter first serial port number : ";
19     //std::cin >> serial_port;
20
21     // Run threads.
22     std::thread sendThread(&SendThreadFunc);
23
24     // Wait threads.
25     if (sendThread.joinable())
26         sendThread.join();
27
28     return 1;
29 }

```

On the converter side, you should receive the following message:



Device Advanced Configuration

Device Information
Device Type : I-7580
Firmware Version : V1.1

Information Communication Log Comport Test Diagnostic Msg.

Parameters

Format : Binary	End char of input data : NONE	Reset
Baudrate : 115200	Diagnosis of time out : None	
Line Control : n, 8, 1	Input fixed length : Disable	
Unit of timeout : 1 ms	timeout value(0~255) : 0	Fixed length : 0

Send

Send String with : CR+LF

AA A0 00 00 00 00 00 00 18 62 01 01 01 01 01 01 00 00 02 00 00 00

Send

Count 1

Clear

Receive

AA A0 00 00 00 00 00 00 18 62 00 01 00 01 00 01 0A 00 00 00 C8 00 00 00

Count 1

Update

Clear

8) You can also track all traffic between the converter and PC

Device Advanced Configuration

Device Information

Device Type : I-7580
Firmware Version : V1.1

Information Communication Log Comport Test Diagnostic Msg.

Format : Binary Message Counts : 2 Save

#	Timestamp(ms)	Tx/Rx	Length	Data
1	9155642	Rx	24	AA A0 00 00 00 00 00 00 18 62 00 01 00 01 00 01 0A 0...
2	8815402	Tx	24	AA A0 00 00 00 00 00 00 18 62 01 01 01 01 01 01 01 0...

< >

Update Clear