User guide

UART

Version 1.0 • June 20 , 2021

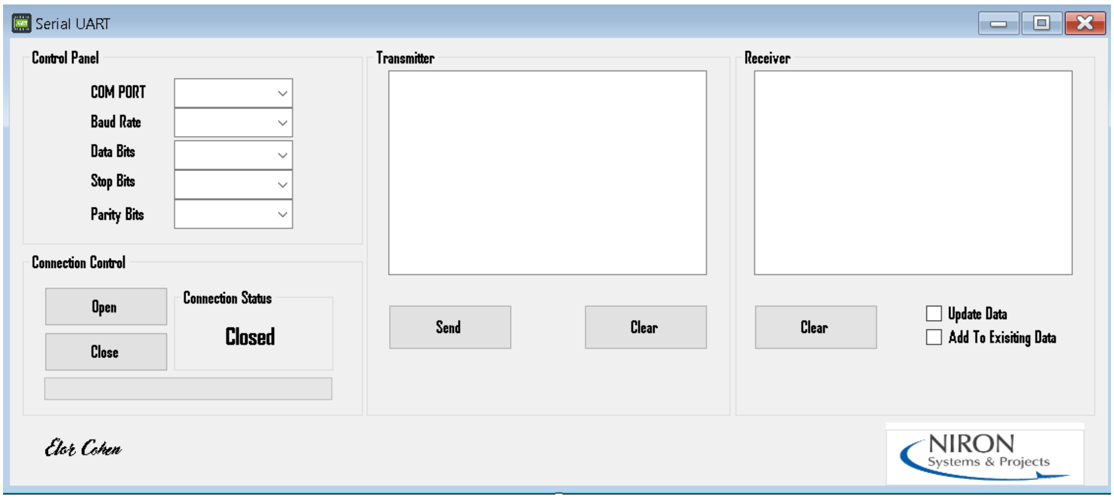
Elor Cohen

1.0 General

This project is a UART interface using a basic GUI for sending and receiving data from/to the UART. The application has been prepared using C#. It is compatible with both Windows 10 and Windows 7.

2.0 System overview

In this section, you will learn how the system works. But first, let's have a look at how the system is built:



Connection Status
Closed

□ Update Data
□ Add To Exisiting Data

There are 4 main components – control panel, connection control, transmitter, and a receiver.

Control panel -

The control panel is responsible for transmitting and has 5 necessary options:

COM PORT- Choose on which port you are willing to send/receive.

Baud Rate – Choose the baud rate(number of symbols per second). There are 8 options, which start from 1200 and up to 115200, The default is 9600.

Data bits – How many bits the data has, there are 4 options, from 4 up to 8 bits.

Stop bits – Synchronization bits that determine where the message is ended. You can choose, one or two, the default is 1.

Parity bits – Low-level error checking, you can choose between none, even, and odd.

Connection control-

The connection control is responsible to open/close the connection between two ends of the system.

Transmitter –

Text box to enter your message, a Send button to send the message, and a Clear button to clear the text.

Receiver –

Text box to display the received data, a Clear button to clear the text, and two checkboxes to let the user decide: overwrite the incoming data or to add up to the existing data.

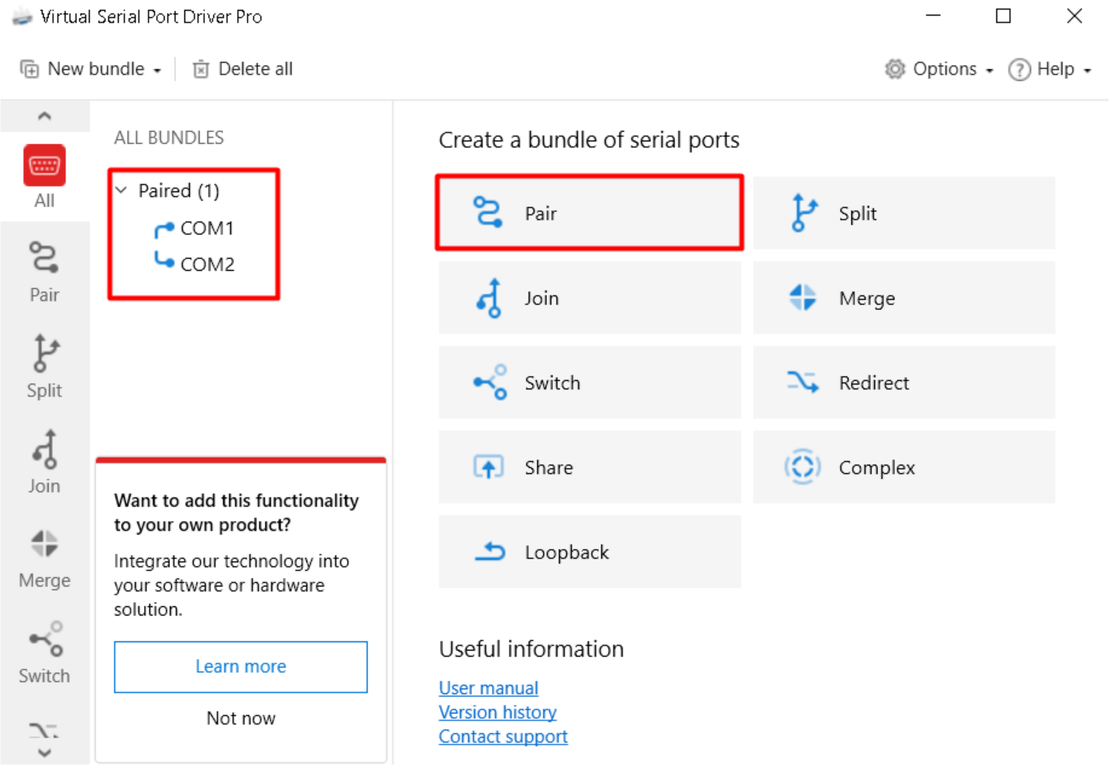
3.0 Activating the system

3.1 First-time users

If you are new to the system, there are pre-installing manners :

- Firstly you will need to download any ***Virtual Serial Port Emulator***. The software emulates serial port functionality connected by virtual null-modem cable in such a way that the system does not see the difference between virtual and real hardware ports.

- When the emulator is on, click on "pair" to pair between two ports, usually they are called "COM1", "COM2", etc...

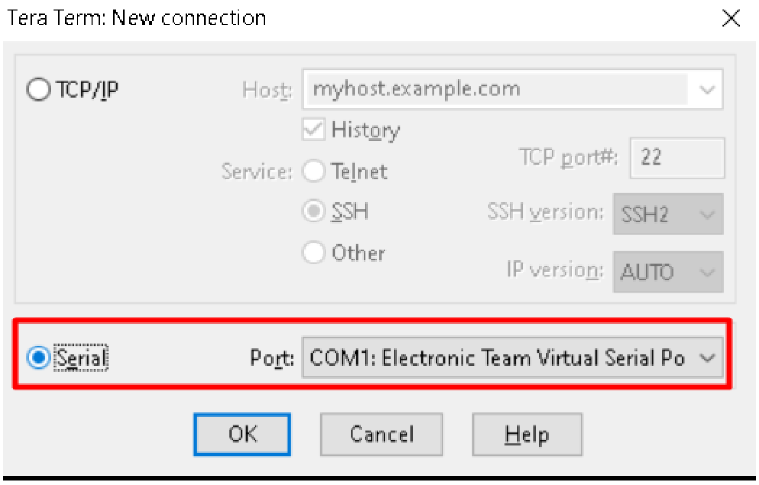


Want to add this functionality to your own product?

Integrate our technology into your software or hardware solution.

Next, you'll be needed to download a terminal such as Tera-Term/Putty.

After the terminal is running, choose "Serial" as your new connection as follows :



Tera Term: New connection

○ TCP/IP Host: myhost.example.com

☑ History

Service: ○ Telnet ● SSH ○ Other

TCP port#: 22

SSH version: SSH2

IP version: AUTO

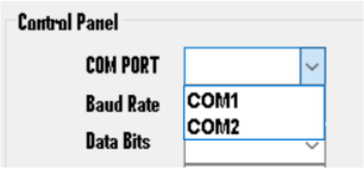
● Serial Port: COM1: Electronic Team Virtual Serial Po

In addition, don’t forget to choose a port, and hit "OK".

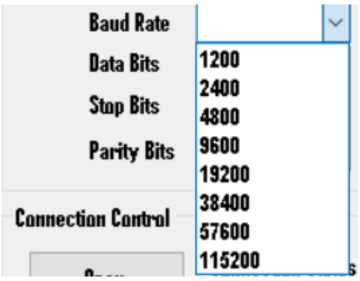
3.2 Starting the UART

If you followed the instructions above, you are ready to activate the UART:

3.2.1 First choose the port you desire, noted that you need to choose the opposite port you choosed in your terminal. For example, if you choosed COM1 in the terminal, in the UART you will be needed to choose COM2. (The author assumes that there are only 2 ports available if not so, make sure not to choose the same port on both ends!)

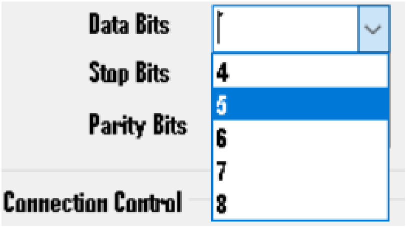


3.2.2 Now choose the Baud rate, the Baud rate is the rate at which information is transferred in a communication channel. In a serial port context, 9600 Baud rate means the port is capable of transferring a maximum of 9600 bits per second.

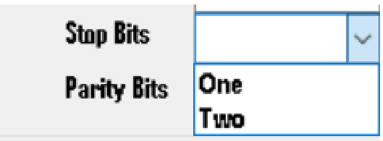


D

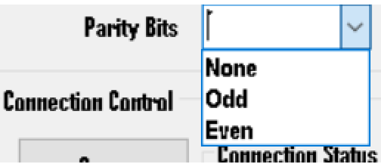
3.2.3 Now choose the number of data bits, simply choose the length of the data.



3.2.4 Now choose the number of stop bits, stop bits are part of synchronization bits (Start bit and Stop bits). These bits mark the beginning and the end of a packet. Because there is only 1 start bit, no selection is needed. Therefore the choice is only for the stop bit and is either one or two.



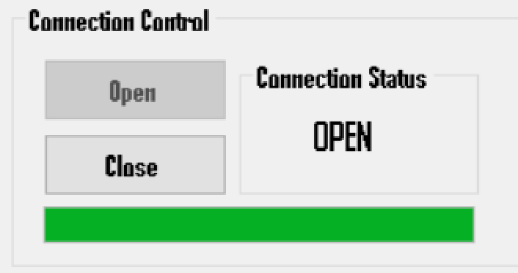
3.2.5 Then choose the Parity bits, Parity bits are for error checking and the choice is either none, odd or even. If you choose **none** that means no parity bit will be in transmitted data, if you choose **odd**, that means that this bit will be on if the number of 1's in the data is odd, and if you choose **even** in the same context, will mean the number of 1's is even.



4.0 Transmitting Data

Now, after you make all the necessary adjustments, you are ready to transmit your data.

First, click "Open", if the connection has succeeded the gauge will turn green, and the connection status will indicate that the connection is open, as follows :

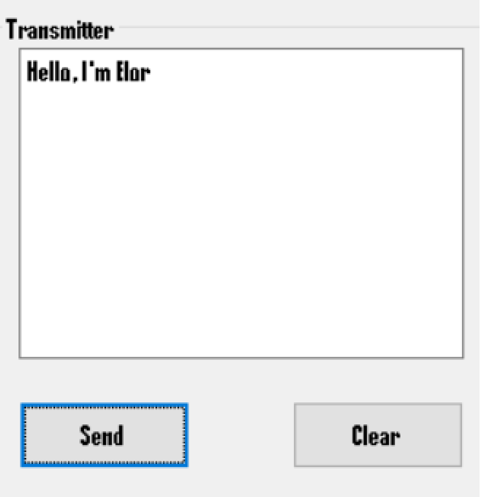


Connection Status
OPEN

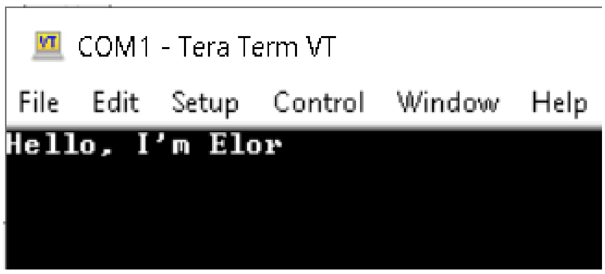
**** Please make sure you follow all the steps, transmitting data will not be possible unless you open a connection. Entering a message and hit "Send" without an open connection will result in an unsuccessful transmit!**

Now, simply enter in the text box of the "Transmitter" the message you want and hit "Send".

Your message should be displayed in the terminal:



Hello, I'm Elor



```
Hello, I'm Elor
```

Cleaning your text-box by simply hit on the "Clear" button.

5.0 Receiving Data

When receiving data, two options are available for you :

Overwrite Data – Choosing this option means that every character received will overwrite the previous character.

Add to existing data – Choosing this option means that every received character will add up to the previous character.

Now, to receive data open the UART and follow the steps above in section 3.2.

Open the connection by pressing the "Open" button. After the connection has been established, data can be received.

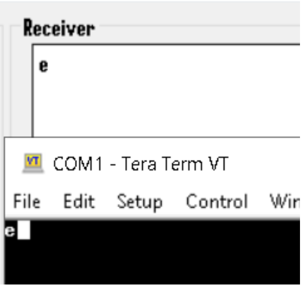
Then choose the form of receiving data: Overwrite or Add up.

Open your terminal and write your message, it should be displayed in the Receiver's text-box.

5.1 Receiving data when Overwrite Data is selected

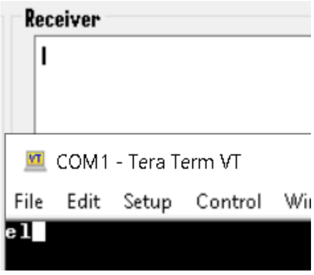
When Overwrite is checked, every character received will overwrite the previous character.

The next shows the receiving process while Overwrite is checked :



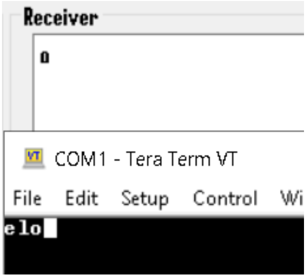
```
e
```

```
e
```



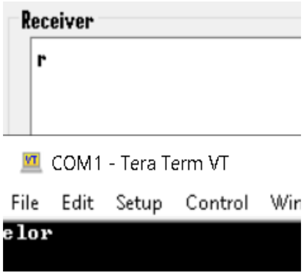
```
l
```

```
el
```



```
o
```

```
elo
```



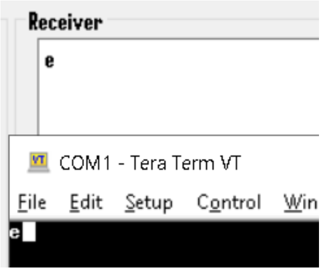
```
r
```

```
elor
```

5.2 Receiving data when Add To Existing Data data is selected

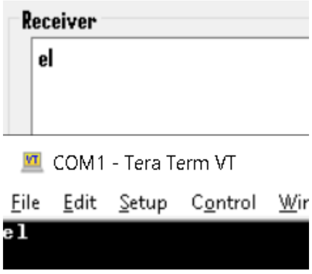
When Add To Existing Data is checked, every received character will add up to the previous character.

The next shows the receiving process while Add To Existing is checked :



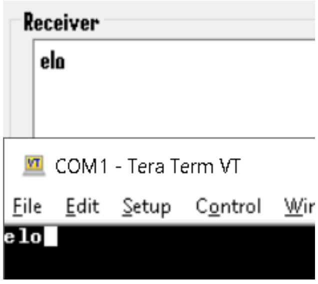
Receiver: e

```
e
```



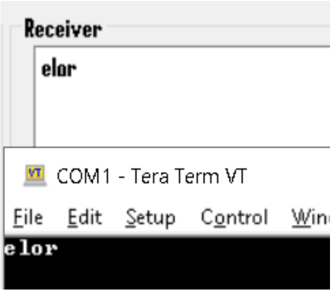
Receiver: el

```
el
```



Receiver: elo

```
elo
```



Receiver: elor

```
elor
```

Cleaning your text-box by simply hit on the "Clear" button.