RS232 PicoScope

Remote Lab Instructions

v3

Created By:	Danie Stoop, Deon Reynders	Date:	02/08/14
Reviewed By:	Steve Steyn	Date:	10/11/14
Reviewed By:	Danie Stoop	Date:	21/02/2015
Reviewed By:	James T	Date:	09/06/2020

Remote Lab Instructions
RS232 PicoScope

Getting Started

- Logon to Electromeet
- The software is installed on the remote labs – search for "picoscope" or "RS232" using ctrl+f

Hardware:

As you will be doing this exercise by way of a remote lab please note that all the hardware is already set up and connected to the relevant remote labs. **Other remote labs will not have the correct hardware setup.**

Software & Hardware used:

The lab was set up with the following hardware:

- Laptop with at two USB ports
- 1x MOXA UPort 1110 USB-to-serial (RS-232) adapter
- 1x 9-pin D-type female to 25-pin D-type male adapter plug ('straight') (DB-9F/DB-25M)
- 1x breakout box
- 1x PicoScope 2204 USB oscilloscope with probes

The following software has been installed:

- RealTerm: Serial Capture Program (for sending characters via the COM port)
- PicoScope 6 software

RS232

The objective of this exercise is to give you an understanding of how asynchronous serial data communications operate, using a breakout box on the COM port of a laptop computer. Since most laptops nowadays do not have physical COM ports, we will use a USB-to-Serial converter to accomplish this. The MOXA adapter used actually has on on-board UART (Universal Asynchronous receiver Transmitter), so the UART has simply been relocated from the motherboard to the USB adapter. The output of the serial port will be monitored by means of a PC-based oscilloscope.
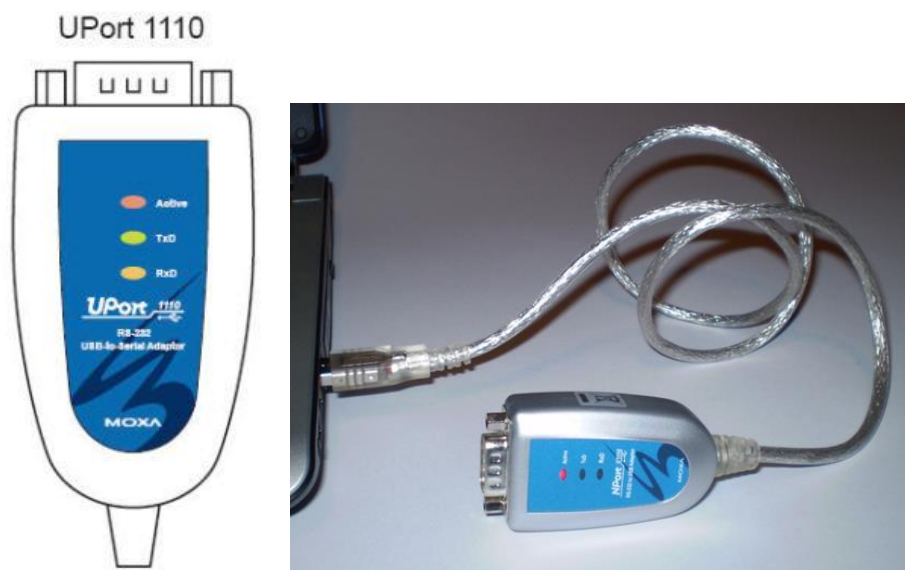
1.0 Implementation

1.1 Basic port setup

As you will be doing this exercise by way of a remote lab, all of the equipment will already be set up, but please read the instructions below to familiarize yourself with the procedure.

Despite the fact that the drivers have already been installed, it is important that you verify the COM port allocated to the UPort 1110.

The USB to serial RS232 hardware being used is shown in the images below:

The purpose of the three LEDs are as follows:

Name	Description
Active	This LED indicates normal operation. If the driver is installed correctly and the adaptor is plugged into a functioning USB port, the Active LED will light up and remain on. If the LED does not light up, there may be a problem with the adaptor, the driver installation, or PC configuration.
TxD	This LED blinks when the adaptor transmits data through its COM port.
RxD	This LED blinks when the adaptor receives data through its COM port.

If the red LED on the converter does not light up, or a yellow exclamation mark shows up against the UPort 1110 entries in the Device Manager, it could be that the driver settings have been altered.

Since this is a remote setup, assume that the LED lights up.

You can confirm the COM port to which the serial device is mapped, by following the below steps:
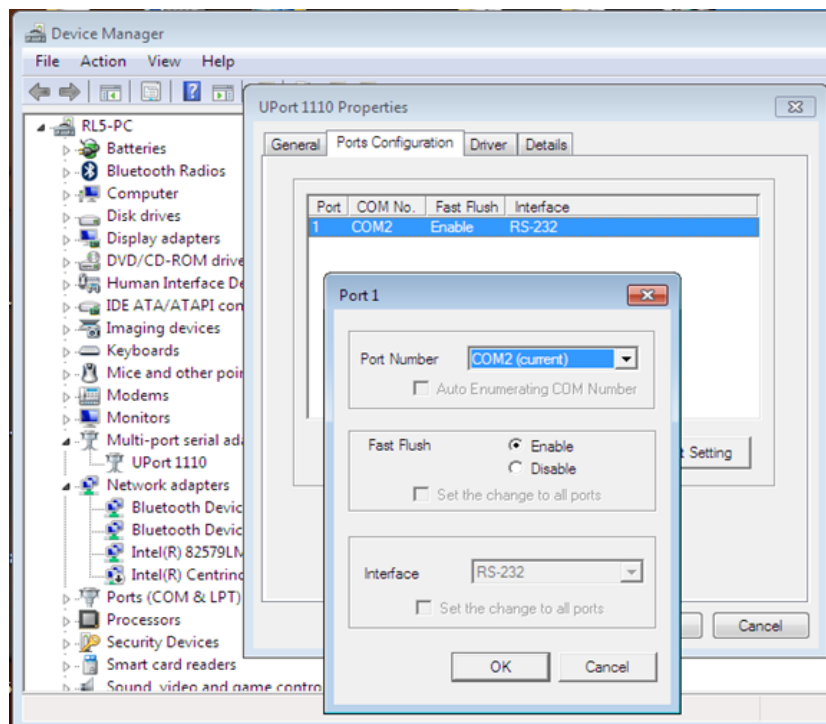Click: Start→Control Panel→System→Hardware→Device Manager.
Locate the entry that reads: 'Multi-port serial adapters' and expand it to show the UPort 1110.

Double-click (or right-click→Properties) on 'UPort 1110' to obtain its properties window.


Select the 'Ports Configuration' tab and note the COM port that has been allocated to the UPort 1110.

PLEASE DO NOT CHANGE IT

(Should a user wish to change to another port number, this can be accomplished by clicking on 'Port Setting' and selecting an appropriate one. On a machine without any COM ports any one could be selected, but on machines with an existing COM ports (typically COM1) you have to select a different one to avoid conflicts)

As an additional check, go back to the Device Manager and locate 'MOXA USB Serial Port (COM2)' (this is the COM port number we selected earlier). Double-click (or right-click→Properties) on the MOXA entry to obtain the MOXA serial port properties. Select the 'Port Settings' tab and check the Universal Asynchronous Receiver Transmitter (UART) settings. These settings can, however, be overwritten by application programs such as RealTerm: Serial Capture Program.
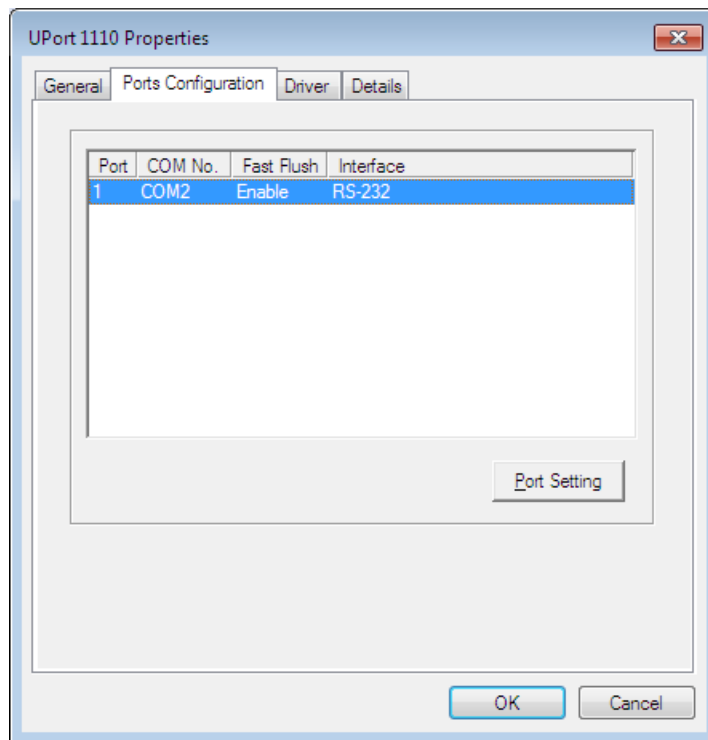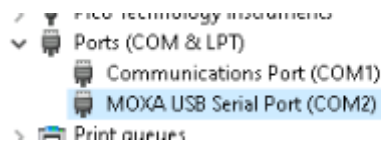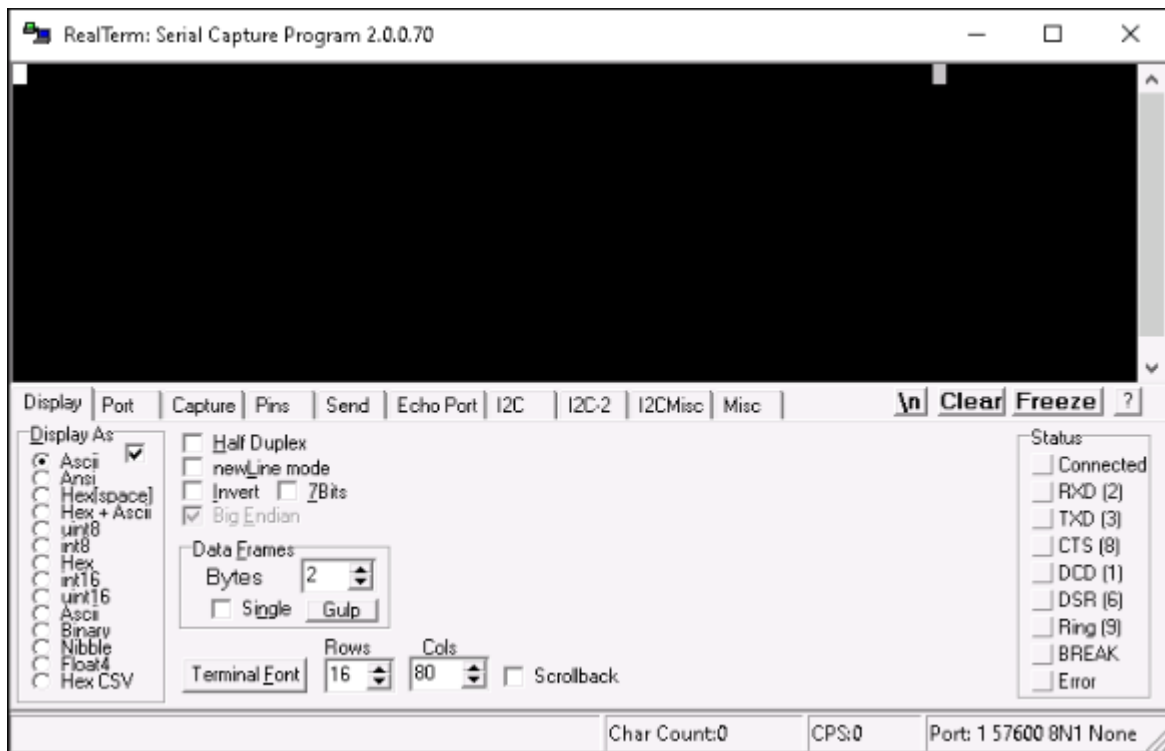

The Moxa 1110 settings are now complete (this should already be done on the remote lab computer so no changes should need to be made)

1.2 Prepare to send characters

1. Run RealTerm: Serial Capture Program by clicking on the desktop icon or on the taskbar at the bottom of the screen.

The software will open up:

2. Click the "Port" tab.

3. Then set the UART parameters for the appropriate COM port (2 in this case). Set the parameters exactly as per the following figure: Baud = 9600, Parity = Odd, Data Bits = 7, Stop Bits = 1, Hardware Flow Control = None.
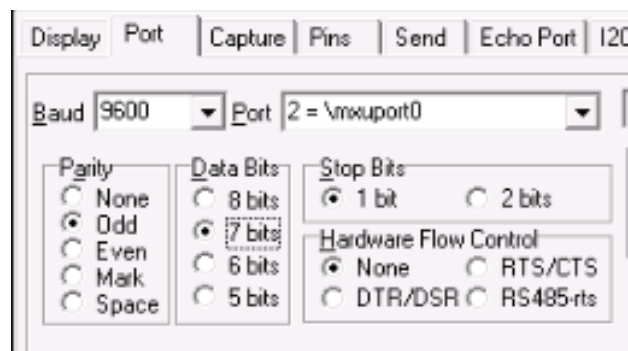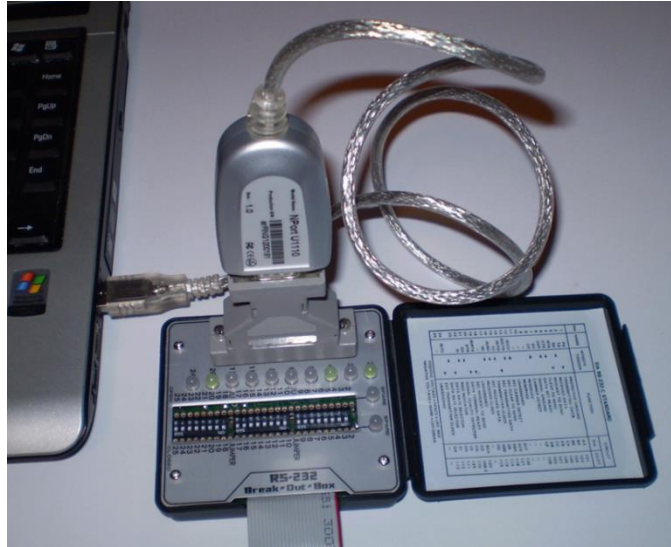

4. Next, start the port monitoring process by clicking "Open" and "Change":

RealTerm is now running and the UART on the Moxa device is configured correctly. Now you must confirm the idle voltage.

1.3 Confirmation of idle voltages

The USB converter is connected to the breakout box by means of the DB-9/DB-25 plug. It looks like this:

Normally in a hands-on situation you would do this by observing the LEDs on the breakout box or by measuring the voltage across the two pins (TxD on pin 2, gnd on pin 7) by means of a voltmeter.

However as we are doing this remotely we can measure the TxD line idle voltage with the PicoScope, *which is a remote configurable oscilloscope, already connected to the correct pins.*

To do this you must:

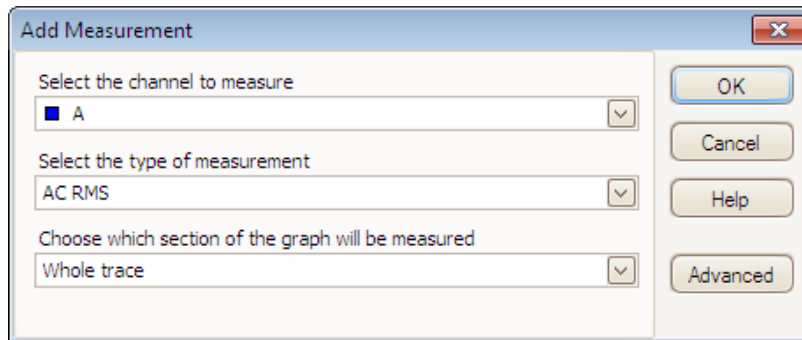5. Run the Picosope6 software. Double-click the PicoScope6 icon on the desktop of the remote computer.

6. We can add a measurement function to the PicoScope, which will allow us to obtain a DC voltage reading. Firstly click "Measurements" at the top of the PicoScope window
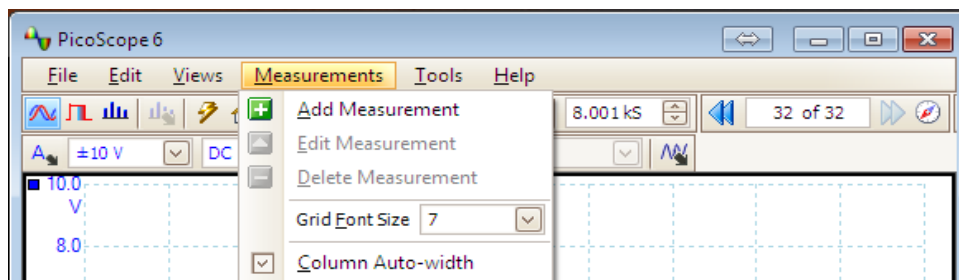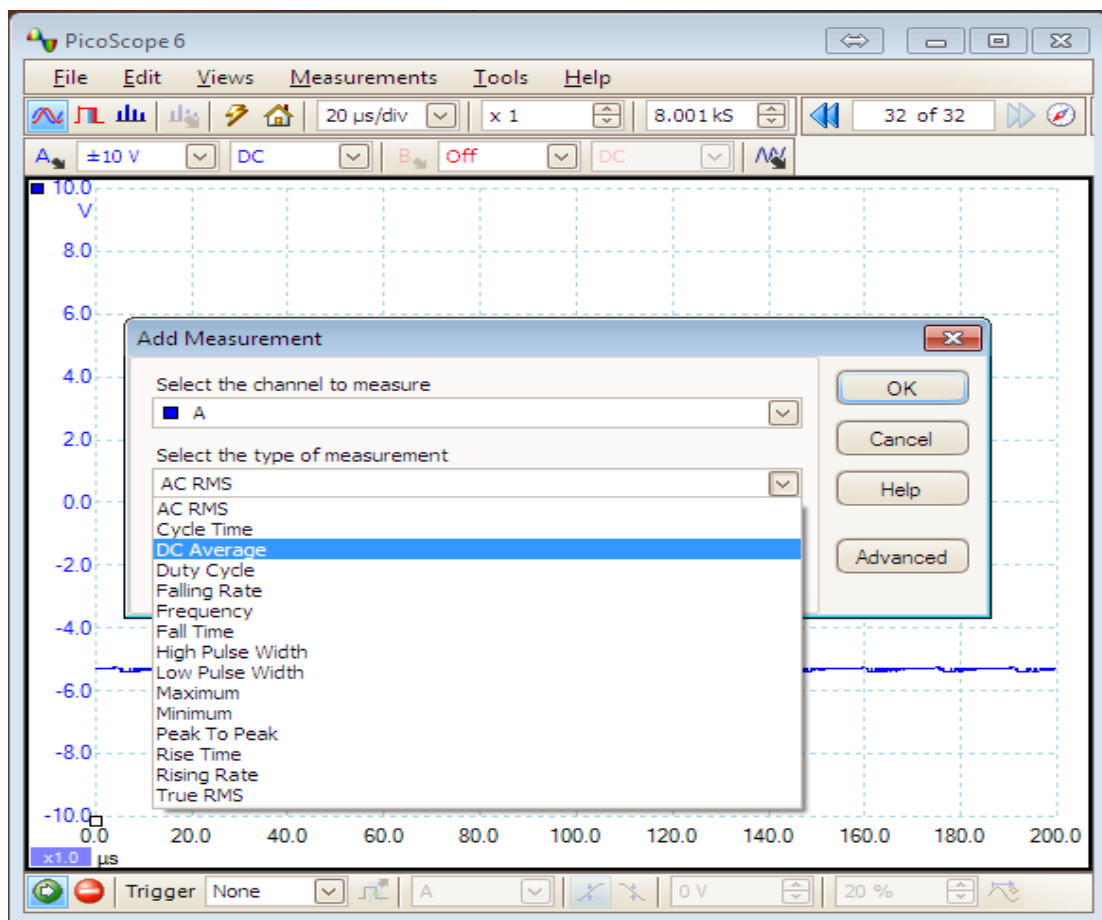
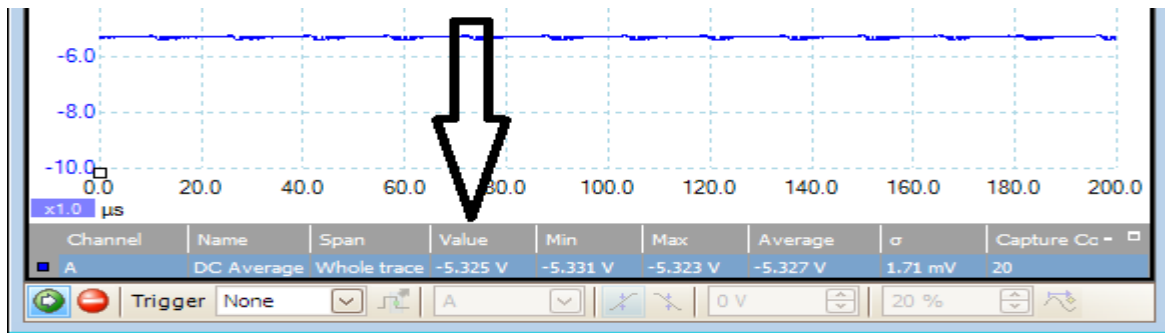

7. Click "Add Measurement"

8. This will bring up the "Add Measure Window"

9. From the Type of measurement selection field select "DC Average"

10. Then Click OK. This will add the measuring function to the bottom of the PicoScope

11. You can then read the average DC voltage level.

Question: What is the average DC voltage on the TxD line when it is idling? State the answer in your assignment (rounded to 2 decimal places).

Instruction: Take a screenshot of this and paste it into your assignment.

1.4 Transmitting a single character and then capturing it on the PicoScope

You now must transmit a single character and capture the output of the TxD line on the PicoScope.

NOTE: THIS IS A TRAP FOR THE UNWARY. The Start/Stop buttons on the PicoScope complement every time you click on them. So, if the PicoScope has stopped capturing, and you click twice on the Start button, you are still in 'stop' mode. However, if you observe the backgrounds of the Stop and

Start buttons you will notice that they change to a blue selection highlight when active.

We already have the PicoScope running but now must change the configuration to capture the waveform. The PicoScope settings should be as follows:

- Trigger: Single trigger, Channel A , rising edge
- X-axis: 200 microseconds per division
- Magnification: X1
- Y-axis full scale: +/-10V
- Coupling: DC
- Tigger level: 0V
- Pre-trigger: 20%

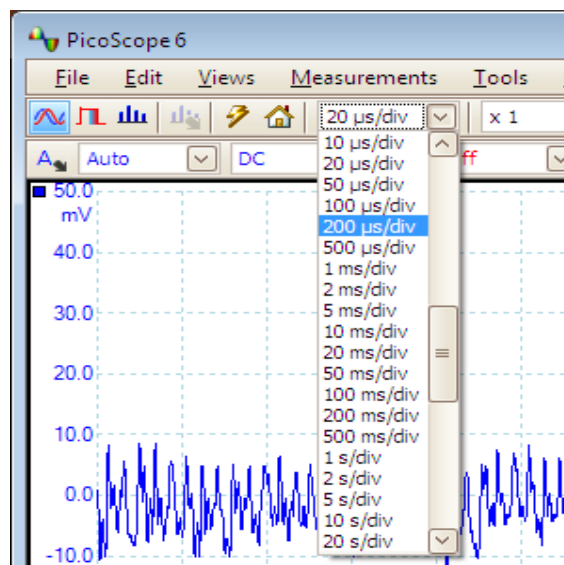Here is how we change to settings

12. On the PicoScope under the trigger Field (at the bottom of the screen) select "Single"
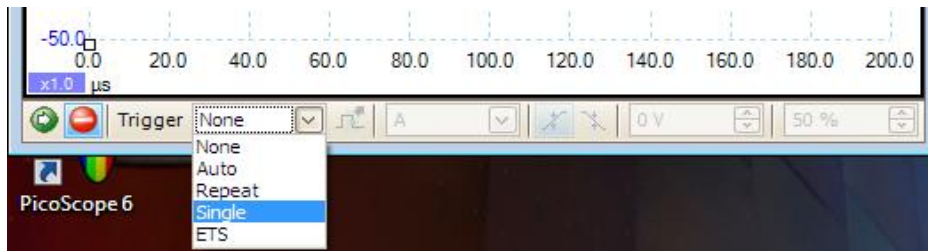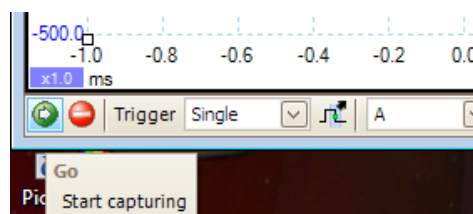
13. Set the pre-trigger to 20%. This defines the point (from left to right across the X axis) where the leading edge of the 'start' bit (on which you are triggering) will appear
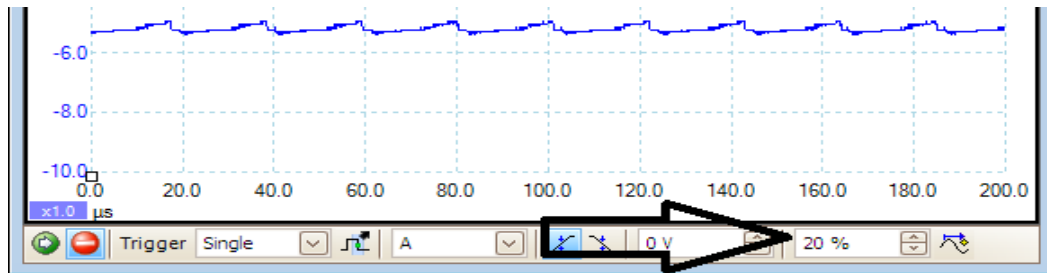
14. At the top of the PicoScope under the Time/div field select 200 µs/div


15. Now click the green "Start Capturing" icon (green Icon at the bottom left of the PicoScope window)

The Picoscope is now ready to capture the character we are about to transmit.

EIT
Engineering Institute of Technology.

16. We already have RealTerm running (it will be in the tool bar at the bottom of the screen if the window is not already on screen). Open the RealTerm window and click the "Display" tab, and select "Half Duplex" – this will output sent values to the black terminal window:
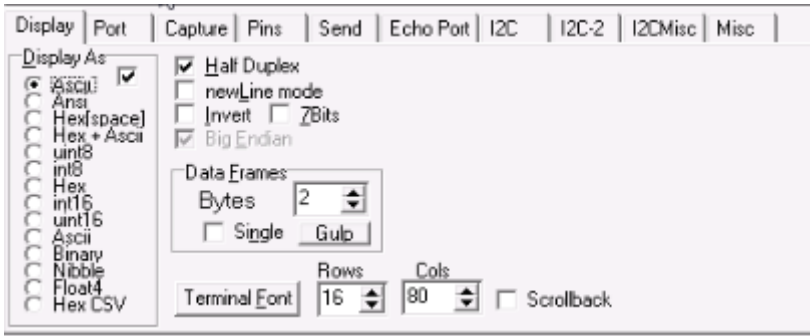
17. On the RealTerm window click on the "Send" tab, enter a value in the field (i.e. "16" or "F", etc.) and click "Send ASCII" – note, the value appears in the black terminal above:
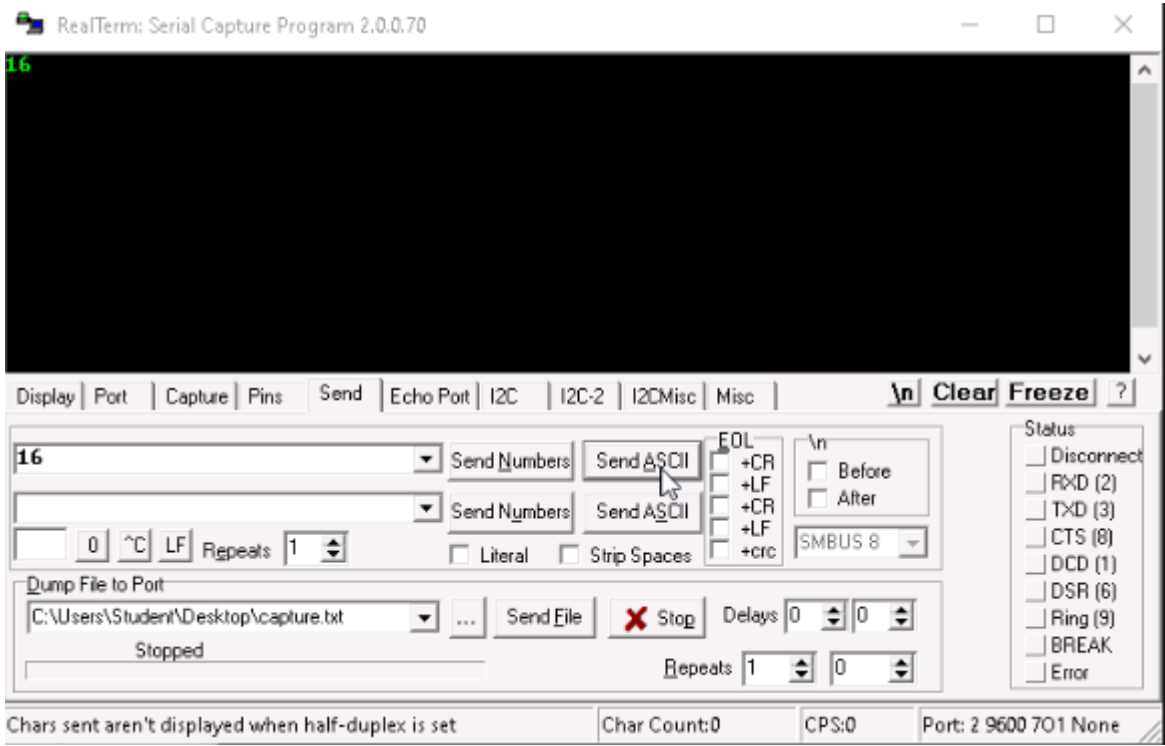
18. The transmission should be captured on the PicoScope now (and the PicoScope capture changed from Running to Stopped), select the PicoScope and observe the waveform (similar to the one below):
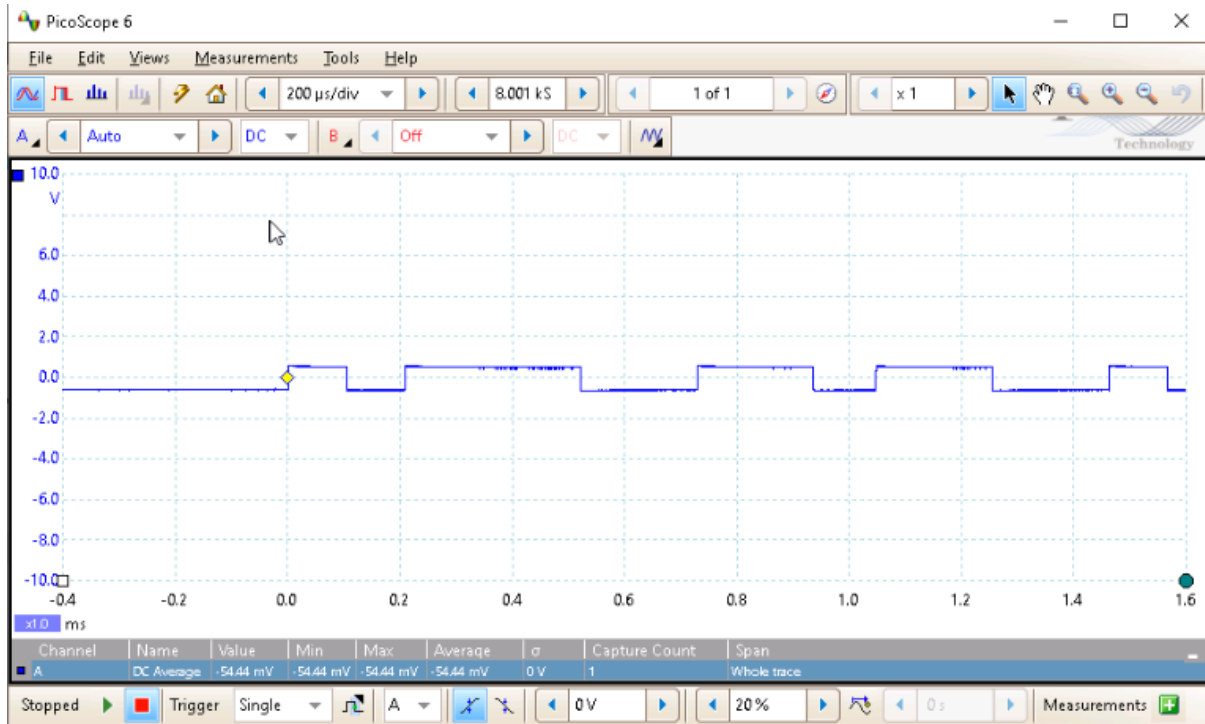
The high and low voltage levels are due to the specific machine to which the converter is connected and are often not perfectly symmetrical with respect to ground.

Instruction: Take a screenshot of this waveform, and paste it into your assignment.

Question: What is the actual bit rate in bits per second? This is also referred to as the baud rate (bit rate = baud rate for RS-232). You do this by first measuring the width of one bit on the oscilloscope (in milliseconds) and then inverting that to get the answer in bps. Show your calculations

Question: What is the effective data rate? Not all the bits in the frame carry information. Calculate the effective data rate by taking your answer in (a) above, and multiplying it with the ratio of actual information bits to total data bits in the frame.

Hint: *For the example above the parity bit is the last 'positive pulse' on the x-axis.*

END OF INSTRUCTIONS