

# Lab 1 – Data Link Protocol First and Second Classes Guide

2025/2026

## First Lab Class

### Objective

Exchange arrays of bytes between two computers using a serial cable and implement the establishment phase of the data link protocol.

### Steps

#### 1. Consider groups of 2 elements per workbench

In NetLab (I322) there are 12 workbenches, numbered from 1-12. Please consider groups of 2 elements per workbench. The group members and the workbench should remain the same across the semester.

#### 2. Select the correct computers

Each workbench has a rack of four computers (tux1 – tux4), and a separate computer over the desk, closer to the centre of the room. In this work, we will use the serial port of this last computer and the one from tux3. To select tux3, you should press Scroll Lock + Scroll Lock + 3 + Enter on the keyboard (workbench 1-6) or press the number 3 on the KVM switch (workbench 7-12). The username and password are written on the workbench desk.

#### 3. Ensure the serial connectivity between computers

To check the connectivity, open GTKTerm on both computers of the workbench and write some characters. The text should appear on the other computer. If this test fails, please check if the selected serial port is the correct one and that the baudrate is the same on both computers. Please report if the problem persists.

There are complete guides in Moodle explaining how the serial port works and how to use GTKTerm.

**Note: the GTKTerm is only used to check the connectivity between the PCs. It must be closed while the data link protocol is running. When you are testing your programs make sure you do not have any GTKTerm instance running.**

#### 4. Use the serial port in non-canonical mode using the example code

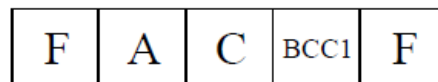
In Moodle, there are two files with example code showing how to write / read data to the serial port in non-canonical mode. These files can be used as a starting point for this project, namely:

- write\_noncanonical.c: This file can be used as the starting point of the transmitter
- read\_noncanonical.c: This file can be used as the starting point of the receiver

You should compile the programs using GCC and run them in the terminal. Note that you should specify the serial port as an argument (/dev/ttySx, being x the index of the serial port and the default value 0). When running the example code, an array with the alphabet (abc...xyz) is sent through the serial port and printed on the receiver terminal.

#### 5. Implement the logic connection establishment

Change the example code to send an array of bytes, defined in hexadecimal (declared as unsigned char), which performs the logic connection establishment. This phase considers the exchange of supervision frames that are composed by 5 bytes, starting with a flag, followed by an address field, a control field, a BCC and ending with a flag.



Each field, defined as unsigned char, has the following values:

**FLAG:**

01111110 (0x7E)

**A (Address):**

00000011 (0x03) – frames sent by the Sender or answers from the Receiver

00000001 (0x01) – frames sent by the Receiver or answers from the Sender

**C (Control):**

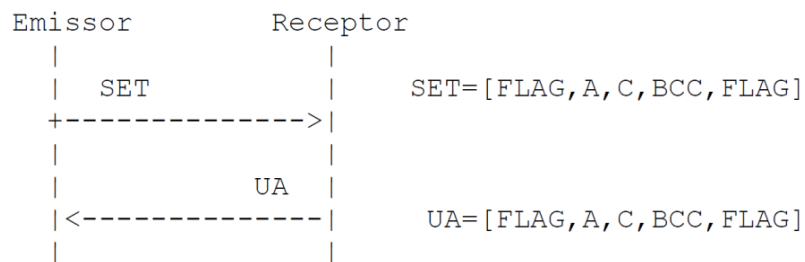
SET: 00000011 (0x03)

UA: 00000111 (0x07)

**BCC1 (Block Check Character):**

XOR of all characters of the header (in this case A and C)

The establishment phase is complete once the transmitter sends a SET frame, the receiver answers with an UA frame after checking that the received frame was correct, and the transmitter correctly detects the UA frame.



For debugging purposes, you can print the hexadecimal values using:

```
printf("var = 0x%02X\n", var);
```

Or, if this does not work due to older versions of the compiler, using:

```
printf("var = 0x%02X\n", (unsigned int)(var & 0xFF));
```

**Please report to the Professor once this step is completed.**