

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

2025/2026

## Second Lab Class

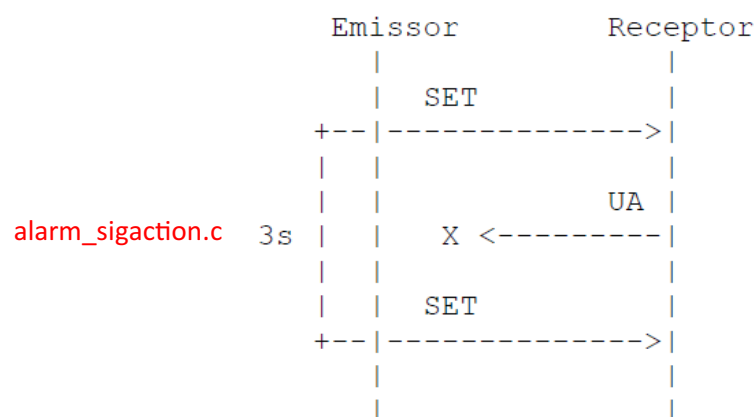
### Objective

Implement the retransmission mechanism and the state machine

### Steps

#### 1. Implement the retransmission mechanism

Using the example code of `alarm_sigaction.c`, implement a retransmission mechanism that retransmits the SET frame after a time-out is triggered (3 s for instance). Each frame can be retransmitted up to a maximum number of times (for instance, 3 times).



The `alarm_sigaction.c` example code is available for download in Moodle. We show some of the most important parts or commands to use:

- Define Alarm Handler:

```
// User-defined function to handle alarms (handler function)
// This function will be called when the alarm is triggered

void alarmHandler(int signal)
{
    // Can be used to change a flag that increases the number of alarms
    alarmEnabled = FALSE;
    alarmCount++;
    printf("Alarm #%d received\n", alarmCount);
}
```

- Configure Alarm Handler:

```
// Install the function signal to be automatically invoked when the timer
// expires, invoking in its turn the user function alarmHandler

struct sigaction act = {0};
act.sa_handler = &alarmHandler;

if (sigaction(SIGALRM, &act, NULL) == -1)
{
    perror("sigaction");
    exit(1);
}
```

- Enable Alarm:

```
// Enable alarm in t seconds
int t = 3;
alarm(t);
```

- Disable Alarm:

```
// Disable pending alarms, if any
alarm(0);
```

The alarm should be disabled once the UA frame is received. Otherwise, the retransmission mechanism will be activated even if the UA frame was successfully received.

## 2. Serial port configuration

The serial port in non-canonical mode can be used to send and receive bytes without the terminal interpreting them. Two important serial port configuration parameters are the VTIME and VMIN, which are explained below:

c\_cc[VTIME]: inter-byte timeout  
c\_cc[VMIN]: minimum number of characters to be read before returning

By modifying these values, different modes can be used:

VMIN	VTIME	Type of read	Effect
0	0	Poling Read	The read() function will return immediately, with either the number of bytes currently available in the receiver buffer, or the number of bytes requested.
> 0	0	Blocking Read	VMIN sets the number of bytes to receive before the read() function returns. As VTIME is zero, the timer is not used.
0	> 0	Read with Timeout	VTIME serves as a timeout value. The read() function will return if a single byte is received, or if VTIME is exceeded ( $t = VTIME * 0.1$ s). If VTIME is exceeded, no byte will be returned.
> 0	> 0	Read with Interbyte Timeout	VTIME serves as an inter-byte timeout. The read() function will return if VMIN characters are received, the time between two bytes exceeds $t = VTIME * 0.1$ s, or the number of requested bytes is received. The timer is restarted every time a byte is received and only becomes active after the first character has been received.

In order to implement the protocol, you should set VTIME and VMIN accordingly. Check the values used in the write\_noncanonical.c and read\_noncanonical.c example codes and try to understand why they were selected.

## 3. Implement the receiver state machine

A state machine should be implemented on the receiver to ensure that frames are received correctly, in order, and without errors. In every situation in which a frame is being received/read, the state machine needs to be used.

We show an example of a state machine to receive the frame SET. This state machine can only process supervision frames. In order to process information frames in future classes, you will need to adapt it.

# State Machine

## Reception of SET message

