

# Vector Faxing System

Marian Wrzesień and Piotr Ryszawa

Industrial Research Institute for Automation and Measurements PIAP, Warsaw, Poland  
{mwrzesien, pryszawa}@piap.pl

**Abstract.** The vector faxing system based on USB fax modems is presented. The system was implemented and made available to users. In the adopted approach, a vector of USB modems does replace solutions based on standalone fax machines, large size of which significantly hampers the construction of a multiple line faxing system in the PSTN environment (Public Switched Telephone Network) – necessary for the effective functioning of a large company. The abstract control model ACM, adopted in the USB hosts which supports analog modem hardware (including USB fax modem devices), is analyzed. It should be emphasized that explicit discussions of the method of mapping telephone lines to – randomly activated during startup of the fax server – USB fax modems, and of the mechanism for assigning user access rights to designated fax modems, are included. Use of the described system increases immunity to threats such as spam, computer viruses, spoofing, or redirecting to fake websites (phishing). The user communicates with the faxing system by the Winprint HylaFax+ Reloaded client.

**Keywords:** USB Fax Modem, multiple line faxing system, CDC Class Communications Equipment, Fax Server.

## 1 Introduction

The aim of research work described in this article was to achieve a vector structure for the faxing system. There are in fact two problems hindering the implementation of the system in this form, which do not occur in case of the single-line, i.e., one sender – one recipient solution. The first problem is the random nature of registration of modems in a computer system after they are plugged into the USB ports. The second problem is lack of a mechanism for assigning users the rights to use particular fax modems.

The first problem was solved by software enforcement of ordering of modems by aliasing their names. The second problem was solved by the use of queue management mechanism in HylaFax+ fax server – the JobControlCmd.

In the article necessary investigations to achieve this objective as well as the complete method are described. The mentioned problems assume that there exists a fax server, the implementation of which was a separate task.

Adoption of the fax communication in large companies that have local area networks and use a large number of telephone lines provides the following advantages:

increased savings and better ecology, increased order in documents, convenient service, reliability, direct connectivity and reduced cabling.

## 2 Implementation of the Fax System

Implementation of the system covered:

- Introduction and setting up the hardware fax server.
- Installation and configuration of the HylaFax+ software server on the hardware server.
- Development and installation of the custom control scripts for the software, for adaptation of the software to the internal requirements of the company.
- Introduction and setting up the private branch exchange to normalize the fax ringing tone.
- Assignment of fax connections to specific fax modems by creating a name table of the aliases of modems.

### 2.1 System Architecture

The system architecture is shown in Fig. 1. Presented here USB fax modems replace conventional fax machines of relatively large dimensions.

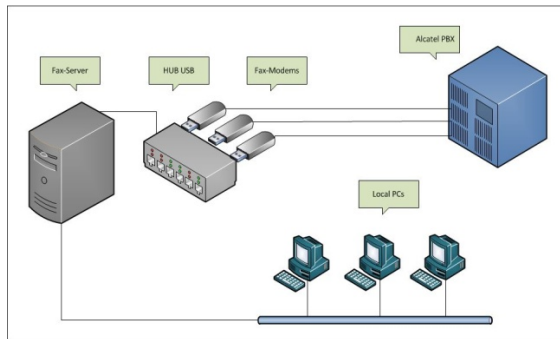


Fig. 1. Faxing system with USB fax-modems

### 2.2 Installing the HylaFAX+ Software Server

**Configuration: Faxsetup-Server.** During the implementation of the faxing system, hylaFax+ and hylaFax+ client were first installed. Installation of the hylafax+ program required prior launch of the faxsetup-server. During execution of the faxsetup, the faxmodem program was launched, whose service faxaddmodem allowed to define and configure USB fax modems with names specified by the user, that is, ttyFax $n$  where  $n \in N$ ,  $N$  – the set of natural numbers. As a result of this operation, a set of

modems anticipated for use in the system was defined. Overall number of defined devices stems from the company needs and is determined by the system administrator. In PIAP 16 fax modems were prepared for use, preserving the possibility of expanding the number to 24.

As a result of execution of the `faxsetup-server` program, a set of files was created, that is the configuration files in the `/etc/hylafax/` directory, executable files in the `/usr/sbin/` directory, and utility files in the `/usr/bin/` directory.

As a part of the presented work the following scripts were developed:

- `10-local.rules` – which at the low level binds USB ports, where modems are attached, to dedicated telephone lines,
- `ttyFaxn.conf` – which initializes and maintains in standby mode the devices `/dev/ttyFaxn`,
- `FaxDispatch` – which specifies the destination address and format of incoming faxes and defines the archiving organization rules for incoming faxes (system of directories, each named after relevant date),
- `FaxNotify` – which specifies the destination address to be notified about events in the faxing system (activity of individual users, the status of delivery of faxes, etc.)
- `StartStopModem` – which starts modems and the `hylafaxd` program,
- `makeFaxDirs` – which creates directories for each device `ttyFaxn`,
- `JobControl` – which contains the mechanism for control of user permissions.

**Daemons for the Faxing System.** The HylaFax+ system processes are managed by daemons: `hfaxd`, `faxq`, `pagesend` and software modems (`faxgetty` with parameter `ttyFaxn`). The `hfaxd` daemon monitors the system and supports the Client-Server process; the `faxq` daemon supports the HylaFax+ planning system; the `pagesend` daemon implements the PET/TAP (Personal Entry Terminal/Telocator Alphanumeric Protocol) and UCP protocol (Universal Computer Protocol) for the HylaFax+. The `faxgetty` program supports the process for supervising fax receiving and fax calls (listening). It uses the `ttyFaxn.conf` configuration file, that ensure continuity of the `/dev/ttyFaxn` modem operation. Faxing system status can be read using `faxstat-s` program. On fax reception, the server calls the `faxcrvd` script.

### 2.3 Access to the HylaFAX System

Access to the fax functionality should be allowed based on user permissions. It should be preceded by registration of users, during which users are given IDs and passwords for subsequent identification. The program `faxadduser` is used for this purpose. User data is stored in the `hylafax/etc/hosts.hfaxd` file.

**Representation of Modems in the Linux USB Host.** USB (Universal Serial Bus) Implementers Forum has established a way to represent the equipment after attaching it to the USB host that complies with the CDC specifications (Communications Device Class) [1]. They also identified a way of interpreting the CDC subclasses of devices that are designed to communicate with the public telecommunications net-

work PSTN. These devices are known as a modem (MODulator-DEModulator), because the data are modulated on the sender side and after reception are demodulated to the original state. The control model for modem devices adopted in USB hosts is the Abstract Control Model (ACM) [2], which is a subclass of CDC and supports analog hardware modems that support the ITU (International Tele-communications Union) V.250 Serial Asynchronous Automatic Dialing and Control – originally called the Hayes standard [3]. It includes a set of commands which are related to the data stream or to a separated control stream via CDC class interface. Currently host micro-controllers are equipped with an embedded system to support the exchange of data via USB. For this area, a standard way of communication is used, which is well supported by any operating system. It meets the CDC/ACM PSTN conditions for cooperation and is presented by the Linux kernel in the form of `/dev/ttyACMn` where  $n \in N$  – supported by the `acm.o` kernel module.

**Creation of Alias Names for Modems.** Attaching each additional USB modem to the server results in creation of a `/dev/ttyACMn` dynamic device by the Linux system [4, 5] (Fig. 2).

```

crw----- 1 uucp dialout 166, 0 07-04 15:47 ttyACM0
crw----- 1 uucp dialout 166, 1 07-04 15:43 ttyACM1
crw----- 1 uucp dialout 166, 2 07-04 15:47 ttyACM2
crw----- 1 uucp dialout 166, 3 07-04 15:43 ttyACM3
crw----- 1 uucp dialout 166, 4 07-04 15:43 ttyACM4
crw-rw---- 1 root dialout 4, 64 07-04 15:41 ttyS0
crw-rw---- 1 root dialout 4, 65 07-04 15:41 ttyS1
crw-rw---- 1 root dialout 4, 66 07-04 15:41 ttyS2
crw-rw---- 1 root dialout 4, 67 07-04 15:41 ttyS3

```

**Fig. 2.** Extract from the Linux OS folder `/dev`

In the described faxing system, each modem is connected to the server via 28-port USB hub. The  $n$ -th `ttyFaxn` modem is connected to the  $n$ -th port of the USB hub, and to the  $n$ -th telephone line. These connections are consistent with the record in the configuration files `config.ttyFaxn`. The server HylaFax+ modems are defined as `ttyFaxn`. These modems are represented in the host by the device: `/dev/ttyACMn`. After attaching fax modems to the server, to each pre-defined fax modem `ttyFaxn` in the system is assigned the `/dev/ttyACMx` device, however, in the general case the indices  $n \neq x$  and this assignment is ambiguous, causing inconsistency between the device `ttyFaxn`, defined in `config.ttyFaxn`, and the phone line physically assigned by the `/dev/ttyACMx` device (they are related to each other in a random fashion). Persistent and unambiguous link of names of specific fax modems to telephone lines can be achieved by assigning the names of USB fax modems to appropriate USB ports. For this purpose is used the “KERNELS” parameter, which clearly defines the location of the modem in the USB hub. This was realized using the `udev` rules Linux functionality. Running the utility `udevadm info-a-n/dev/ttyACM0 | grep KERNELS` it was possible to record features of the device associated with the specific location of the

modem in the USB hub, which was attached in turn to each of the USB ports of the server. For each location of the USB modem, values of the KERNELS parameter were read. Then the file was created /etc/udev/rules.d/10-local.rules, which includes a set of relations between the pre-defined ttyFxn modems and KERNELS parameters obtained during investigations. Presented below is an extract from the aliases file for one modem attached to port No. 1 in the USB hub, which was attached in turn to each of the 6 USB ports available on the server – the USB host.

```
SUBSYSTEMS=="usb", ATTRS{serial}=="24680246", KERNELS=="1-1.2.6", SYMLINK+="ttyFax01"
SUBSYSTEMS=="usb", ATTRS{serial}=="24680246", KERNELS=="1-2.2.6", SYMLINK+="ttyFax01"
SUBSYSTEMS=="usb", ATTRS{serial}=="24680246", KERNELS=="1-3.2.6", SYMLINK+="ttyFax01"
SUBSYSTEMS=="usb", ATTRS{serial}=="24680246", KERNELS=="1-4.2.6", SYMLINK+="ttyFax01"
SUBSYSTEMS=="usb", ATTRS{serial}=="24680246", KERNELS=="1-5.2.6", SYMLINK+="ttyFax01"
SUBSYSTEMS=="usb", ATTRS{serial}=="24680246", KERNELS=="1-6.2.6", SYMLINK+="ttyFax01"
```

Fig. 3. Extract from the 10-local.rules script

Possibility of taking into account relations contained in the above file creates situation, where each of the ttyFxn modems is randomly tied with /dev/ttyACMx device established by the system, but at the same time is rigidly tied to a particular USB port and – thus – to the known number of telephone line (Fig. 4).

```
crw----- 1 uucp dialout 166, 0 07-04 15:47 ttyACM0
crw----- 1 uucp dialout 166, 1 07-04 15:43 ttyACM1
crw----- 1 uucp dialout 166, 2 07-04 15:47 ttyACM2
crw----- 1 uucp dialout 166, 3 07-04 15:43 ttyACM3
crw----- 1 uucp dialout 166, 4 07-04 15:43 ttyACM4
lrwxrwxrwx. 1 root root 7 07-04 15:41 ttyFax01 -> ttyACM2
lrwxrwxrwx. 1 root root 7 07-04 15:41 ttyFax02 -> ttyACM0
lrwxrwxrwx. 1 root root 7 07-04 15:41 ttyFax03 -> ttyACM3
lrwxrwxrwx. 1 root root 7 07-04 15:41 ttyFax04 -> ttyACM1
lrwxrwxrwx. 1 root root 7 07-04 15:41 ttyFax05 -> ttyACM4
crw-rw---- 1 root dialout 4, 64 07-04 15:41 ttyS0
crw-rw---- 1 root dialout 4, 65 07-04 15:41 ttyS1
crw-rw---- 1 root dialout 4, 66 07-04 15:41 ttyS2
crw-rw---- 1 root dialout 4, 67 07-04 15:41 ttyS3
```

Fig. 4. Extract from the Linux OS folder /dev

In PIAP the principle was adopted that every dedicated telephone line is linked to a specific organizational department in which designated employees have access to the fax.

**Assignment of Rights to Faxmodems.** Hylafax+ server does not provide any method for assigning users who are sending faxes to individual faxmodems. In order to solve this problem the queue management mechanism – JobControlCmd – was used. JobControlCmd is a parameter of the configuration file, which specifies the script executed on fax sending event – it is shown in Fig. 5.

```
#!/bin/sh
- /var/spool/hylafax/etc/setup.cache
- /var/spool/hylafax/bin/common-functions
- /var/spool/hylafax/etc/OwnersFaxyKO
QFILE=/var/spool/hylafax/sendq/q$1
parseQfile
case "$modem" in
ttyFax01)
    if [[ ! ${faxyFM[@]} =~ $owner ]]; then
        echo "RejectNotice: \"No rights to the modem $modem.\"";
    fi;;
ttyFax02)
    if [[ ! ${faxyFG[@]} =~ $owner ]]; then
        echo "RejectNotice: \"No rights to the modem $modem.\"";
    fi;;
ttyFax03)
    if [[ ! ${faxyMDS[@]} =~ $owner ]]; then
        echo "RejectNotice: \"No rights to the modem $modem.\"";
    fi;;
    *) echo "RejectNotice: \"Modem is not selected\"";;
esac
exit 0
```

Fig. 5. JobControlCmd script

Operation of this script ensures the fax sent by unauthorized user is rejected and a fax transmitted by the user who is located in the specified database is accepted.

## 2.4 System Startup and Testing

**Testing Fax Modems.** Confirmation of communication with each of the modems ttyFaxn was performed using minicom software. At the initial phase of testing the faxing system signals did not conform to the requirements of the certification standard [6, 7]. To solve this problem the intermediate KX-TDA100D Panasonic PBX was installed between USB modems and the main switchboard – Alcatel PBX. After this modification, control and modem communication function properly [8, 9]. This allowed for running USB fax modems and proceeding with the next step, that is, deployment of the fax system software.

**Testing Communication between the PSTN Fax Modems.** Connectivity and communication between the modems was tested using the sendfax program, a component of the HylaFax+ software. Fax transmission was carried out between the PIAP Alcatel PBX and PIAP Panasonic PBX. After transmission, the realization of the commands specified in the FaxDispatch was examined. On the basis of the results the correctness of the system was stated.

**Testing Communication between the PSTN Fax and Fax-Modems.** For this test was used fax machine connected to the telephone line. While sending fax outside the

PIAP the sendfax command was used. Then with an external fax a message was sent to the recipient's internal number in PIAP. During the transmission, the state of fax machines was monitored using faxstat command. After transmission the realization of the command specified in FaxDispatch was examined. In all cases the correct operation of the system was stated.

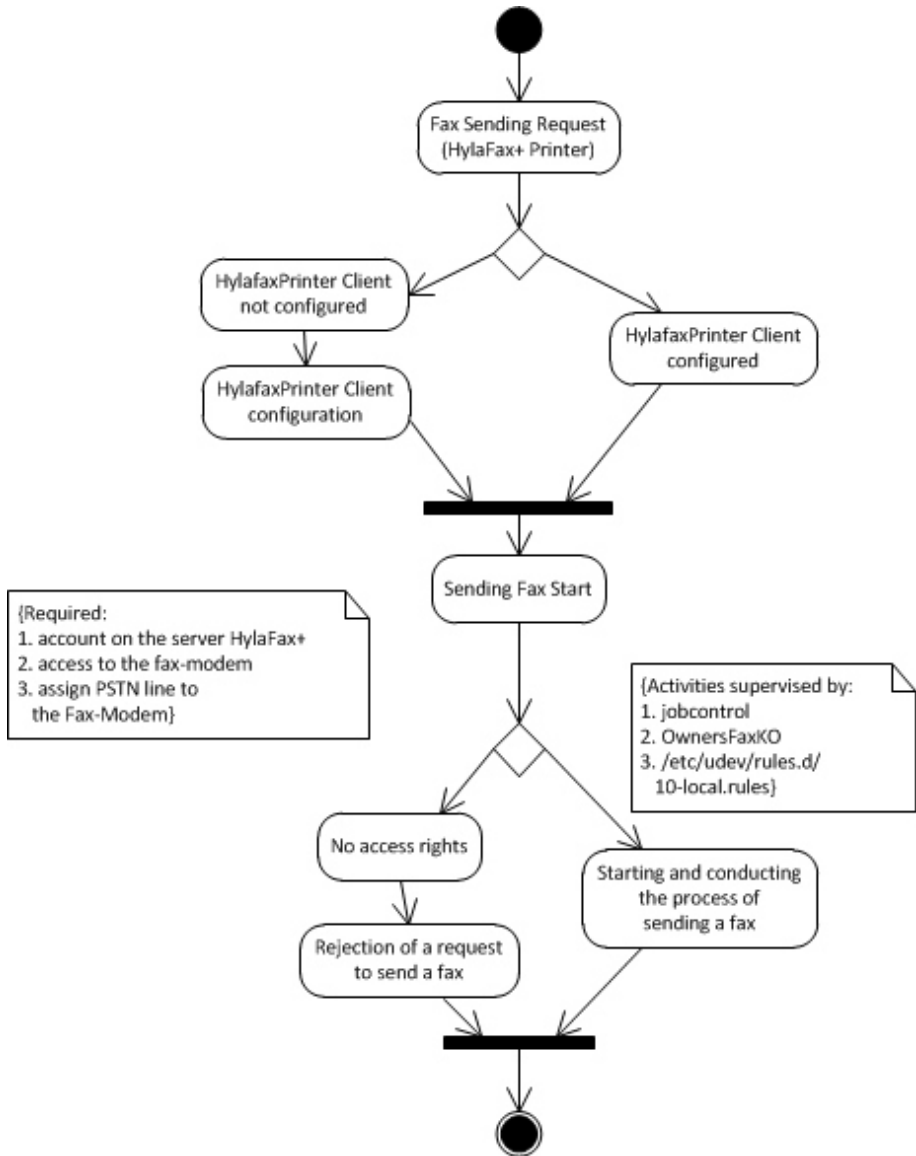


Fig. 6. Sending faxes UML algorithm diagram

### 2.5 Sending Faxes. Implementation of HylaFax+ Client for Windows 7

In order to allow users to send faxes you need to implement the Windows client software. For this purpose, the Open Source Software Winprint HylaFax+ Reloaded was selected of the available programs. The criteria which guided the selection of the software were: compatible with PIAP systems used in OS Windows (32 and 64 bit Windows 7, Windows 8) and intuitive user interface. Software Winprint HylaFax+ Reloaded enables you to use the integrated address book. By default, this book is stored in CSV format at the location specified by the user. There is also possibility to use the Outlook address book using MAPI (Messaging Application Program Interface) and the database which enables connecting using ODBC (Open Data Base Connectivity). Sending of faxes is illustrated in the UML algorithm diagram (Fig. 6).

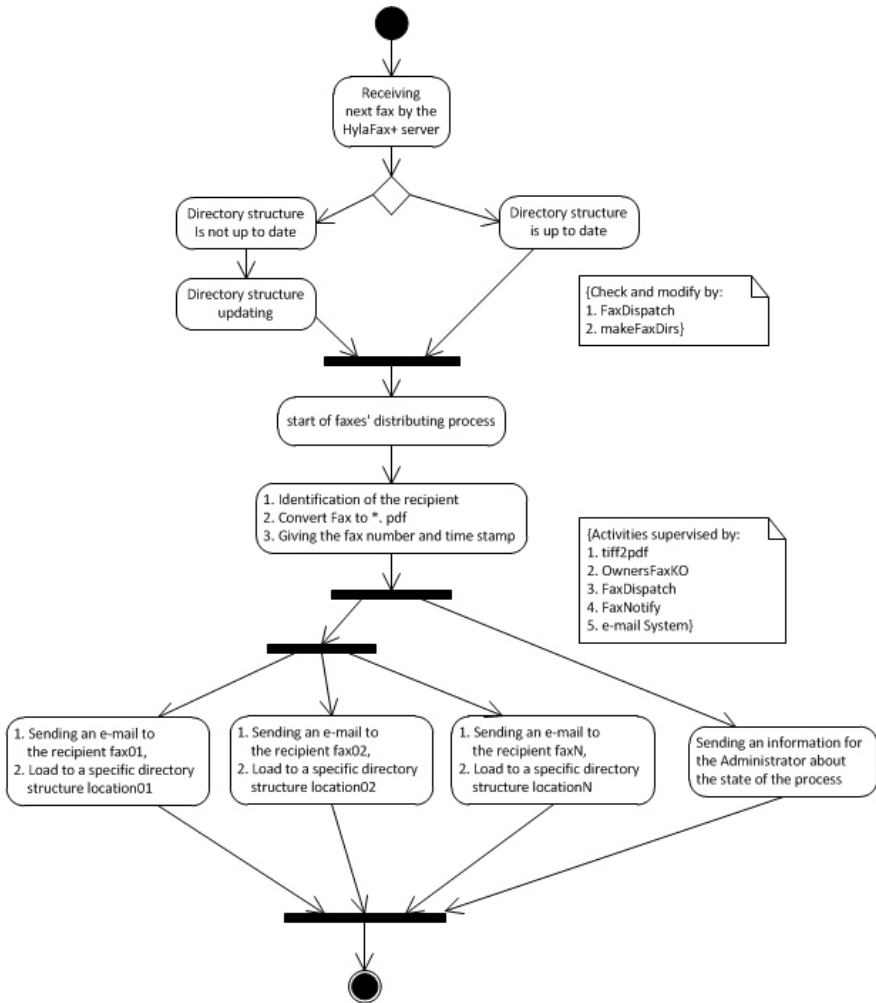


Fig. 7. Receiving faxes UML algorithm diagram



## 2.6 Receiving Faxes. Organization of the Distribution and Archiving of Received Faxes

Received faxes are sent by e-mail to the addressees and stored in the directories associated with each of ttyFxn modems. The list of recipients is defined in the FaxDispatch script file. The ttyFxn is assigned to appropriate e-mail groups of users eligible to receive faxes from the specific fax modem. These groups consist of department workers authorized to use the fax system. All recipients which are assigned to these groups are defined in the file which defines e-mail system users. Receiving faxes is illustrated in the UML algorithm diagram (Fig. 7).

```
# ! /bin/bash
#
. /var/spool/hylafax/etc/OwnersFaxyKO
FaxDirBase="/home/services/Faxy/"
ReceivedFaxesPdf="$FaxDirBase$DEVICE/$(date '+%Y')/
  $(date '+%m')/$(date '+%d')/* .pdf"
Numer=`echo $DEVICE | sed -e 's/ttyFxn//;s/^0//`
FaxyKO=(
  " "
  "faxyFM"
  "faxyFC"
  "faxyMDG"
)
LANG=pl_PL
FILETYPE=pdf
/var/spool/hylafax/bin/makeFaxDirs
/usr/bin/tiff2pdf
  $FILE -j -o $FaxDirBase$DEVICE/$(date '+%Y')/$(date '+%m')/
  $(date '+%d')/`${FaxyKO[$Numer]}$(date '+%Y%m%d%H%M') .pdf
chown uucp:${FaxyKO[$Numer]} $ReceivedFaxesPdf
chmod 750 $ReceivedFaxesPdf
SENDTO=${FaxyKO[$Numer]}@piap.pl
NOTIFY_FAXMASTER=errors
```

Fig. 8. FaxDispatch script for three departments

```
# ! /bin/bash
#
LANG=pl_PL
#
if [ "$WHY" != "done" ]; then
  RETURNFILETYPE=pdf;
  RETURNTECHINFO=yes;
  RETURNTRANSCRIPT=yes;
  NOTIFY_FAXMASTER=faxyNI@piap.pl
fi;
```

Fig. 9. FaxNotify script

When the FaxDispatch is executed, it runs the makeFaxDirs program contained therein that manages the creation of directories that store faxes and supervises access to these directories. Using the script ownership rights are granted to the employees belonging to the group faxyXX, where XX is the department symbol. The received faxes are included into the PIAP global system for electronic documents archiving. The method of notification of the recipients is defined in the script FaxNotify on the HylaFax+ server side.

### 3 Summary

As a result of the conducted work and research, the aim stated in the introduction was achieved. The problem of random representation of modems in the system was solved by means of enforced placement of modems with mechanism of modem names aliasing. The problem of establishing a methodology for assigning to the users rights to send faxes via particular fax modems was solved through the use of queue management mechanism in Hylafax fax server – JobControlCmd. The objective to equip the system with the possibility of separating the fax transmit / receive channel for an individual user, with clearly defined fax modem and telephone line corresponding to them, was achieved. A secondary objective of eliminating the bulky fax machines was also achieved thanks to miniaturized USB fax modems.

### References

1. Gupta, A.: Communication Device Class (CDC) Host2009. Microchip Technology Inc. (2009), <http://www.microchip.com>
2. Teamwork, Universal Serial Bus Class Definitions for Communication Devices, Version Atmel Corp., AT91 USB (January 19, 1999)
3. Teamwork, Serial Asynchronous Automatic Dialing and Control, ANSI/TIA-602, <http://www.eia.org>
4. Teamwork, Universal Serial Bus Specification, version 1.0 and version 1, <http://www.usb.org>
5. Teamwork, Universal Serial Bus Common Class Specification, version 1.0, <http://www.usb.org>
6. Jajszczyk, A.: Wstęp do telekomutacji, WNT, Warszawa (2008) (in Polish)
7. Dziczowska, M., Dziczowski, L.: Obsługa i budowa modemu. Helion, Warszawa (1997) (in Polish)
8. Kula, S.: Systemy teletransmisyjne. WKL, Warszawa (2006) (in Polish)
9. Teamwork, Vademecum Teleinformatyka cz. I, IDG, Warszawa (1999) (in Polish)