

ULAN COMMUNICATION PROTOCOL FOR LABORATORY INSTRUMENTS, HOME AUTOMATION AND FIELD APPLICATIONS

P. Píša P. Smolík

*Department of Control Engineering, Faculty of Electrical
Engineering
Karlovo náměstí 13, 121 35 Praha 2
CZECH REPUBLIC
fax : +420 24357330 and e-mail : pisa@cmp.felk.cvut.cz,
petr.smolik@wo.cz*

Abstract: The uLan protocol is message oriented multi-master communication protocol for microcontrollers. It uses 9-bit character format for addressing and link control and RS-485 physical layer. The protocol is used in HPLC instruments and analyzers for instruments control and data acquisition. The protocol has potential to be used in the home automation applications and some other industrial field applications for its open nature design and proven implementation. The software support is provided for more RS-485 cards, converters for PC RS-232 ports and embedded microcontrollers. One of applications utilizing uLan communication is CHROMuLAN HPLC system software, which provides affordable solution for HPLC assay mathematical data processing and complex HPLC instruments sets control over uLan communication network. Other successful field example utilizing uLan communication is agriculture automatic feeding system.

Keywords: communication, protocol, RS-485, multi-master, HPLC

1. INTRODUCTION

uLan is multi-master message oriented communication protocol which utilizes RS-485 signal levels. Characters are transferred same way as for RS-232 asynchronous transfer except parity bit, which is used to distinguish between data characters and protocol control information and addressing. A physical layer consists of one twisted pair of leads and RS-485 transceivers.

Use of 9-bit character simplifies transfer of binary data and for intelligent controllers can lower the CPU load, because of the CPU need not to care about data characters send to other node.

One of the problems of 9-bit communications is missing standardization of message protocol.

Drivers and formats of one possible implementation of uLan message protocol fills this open area.

uLan protocol is used for long time in the modern HPLC systems produced by INGOS company already and its proven design and features made it to extend to the other areas as well. It has been used in medical instruments (infusion pumps), engine control units, automatic feeding system in agriculture and its finding its way into home automation systems now.

2. LITTLE ABOUT ULAN HISTORY

The design of the protocol has been sketched first in 1992 during development of new modular HPLC system for Laboratory Instruments com-

pany. The system has been designed as set of autonomous instruments (spectrophotometer and pump at beginning) which could operate separately with local gradient program and classical detector analog output. But mutual digital communication and digital interconnection with a PC computer for data acquisition and chromatographic assay control has been bear in mind from beginning. The pump design has been build upon digital feedback control of DC motor. The spectrophotometric detector has employed high resolution in house made AD converter and fully digital signal processing for absorbance computation which was revolutionary at that time.

The main design criteria of communication protocol have been multi master capability, simple physical layer, deterministic media access control, cheap connection to PC computer and low processing power requirements for already fairly loaded microcontrollers in the instruments.

The RS-485 physical layer has been selected for its good noise immunity, bus topology, minimal wires count (only 2 wires) and possibility to be galvanically isolated. The selection proved as good choice which win even for other industrial communication technologies, Profibus for example.

Use of 9-bit character format simplifies transfer of binary data and for intelligent controllers can lower the CPU load, because of the CPU need not to care about data characters send to other node. Many of todays microcontrollers implement 9-bit extension in their UART (all Intel 8051 and 8096 based MCUs with UART, members of Motorola 683xx family - 68376, Hitachi H8 microcontrollers, TI msp430 to name some of them). Intel has developed a multi-protocol UART i82510, which is very well suited for implementing 9-bit communication interface for PC computers. The second example of the chip, which is well suited for 9-bit communication, is OX16C954-PCI produced by Oxford Semiconductors. A converter and some software workaround for classical PC serial port controllers has been designed from the first uLan day as well.

The protocol design has been so capable that base frame format and data exchange pattern has not require revision over last decade. Even object oriented higher layer has been planned from beginning but it has been implemented little later in 1995 when the uLan protocol finds it way into natural gas combustion engine control unit. It has been used there for data acquisition and the engine parameters tuning. The PC side of the software has evolved and changed over the time. The first DOS based implementation of the protocol has been used in the Turbo Visions based development software. Than it has been used for infusion systems monitoring and batch parameters trans-

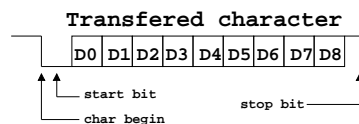


Fig. 1. 9-bit Character Format

fers. The uLan has replaced discrete I/O control in amino-acids analyzers control software for DOS called Micro. The protocol implementation has been rewritten as independent full featured driver. It has been developed on Linux system then it has been extended by OS independence layer to support Windows NT kernel mode driver (KMD). The Windows 2000/XP WMD driver model has been added later. The USB to uLan converter support is in development now.

3. ULAN MESSAGE PROTOCOL

3.1 Message Frames Format

The data frame is a basic communication unit of the uLan protocol. The frame has its destination (node address, general address or not addressed reply start), source node, frame type or command, end mark and integrity check xor_sum. The frame consists of sequence of 9-bit characters. Characters are transferred asynchronously, so every character has one start bit, nine data bits and one stop bit, see fig 1. Total transfer time of one character is equal to transfer of 11 bits. Control characters are transferred with bit D8 equal to one. These control characters appears only on begin and end of the data frame.

The data frame starts with a control character equal to the destination node address (DA_{adr}) or an indication of the reply frame (uL_Beg). The receiving of this character wakes up all nodes and every node decides if rest of the frame will be received. The second character in the frame is an address of the source node (SA_{adr}) and next character is command number or message type (Com). These characters and continuing frame data body are transferred with D8 equal to zero. No length information is sent with frame and frame data body can contain from none to maximum data characters defined by maximal admissible blocking time for other nodes. It is recommended, that data length should not exceed 2kB. The end of data body is marked by the frame end control character, which is followed by the integrity check code (xor_sum). The check code (xor_sum) is counted from all control and data characters of the frame as cumulative xor and addition of one.

The frame end control character selects, what should a node do after receive of the frame. There are four possible values (uL_End, uL_Arq, uL_Prq, uL_Aap).

Data frame format

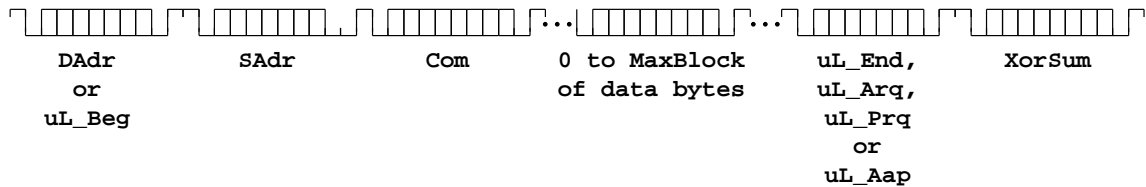


Fig. 2. uLan Frame Format

uL_End frame should be transferred into the input queue

uL_Arq frame should be transferred into an input queue, but the sending node is waiting for the acknowledge about successful receiving of the frame. The frame is acknowledged by sending of the character (uL_ACK). In case of xor_sum error or other undefined problem negative acknowledge should be sent (uL_NAK). Delay equivalent to transfer time of more than three characters is considered by the sender as negative acknowledge or missing node with specified address too. The input queue full condition can be signaled to the sender by “wait please” negative acknowledge (uL_WAK). It solves the problem of overloading of the network by repeating attempts to deliver a message to the node, which has no room to receive messages at the moment. The sender node can wait or process other messages before next attempts. uL_Arq cannot be specified for frames with general destination address and reply frames, because of it can lead to parallel sending of uL_ACK from more nodes, which leads to line collision.

uL_Prq proceed request marks frame that needs immediate processing after reception. Next activity is fully defined by command of the frame (Com). Numerical values of these command should be greater than 80h. Processing of such frame can lead to sending or receiving of next frame (for example memory read and write commands implied to use next frame for data).

uL_Aap same as uL_Prq, but acknowledge should be sent before a start of the frame processing. uL_NAK should be sent in case, that the command is unknown to the receiver node.

The generic uLan data frame format is shown in the Figure 2. A complete description of uLan frames and all protocols details is provided in (1).

3.2 Access Arbitration and Timing

The RS-485 line has no standard instruments to distinguish an access collision. The collision can be found only by receiving of corrupted message with bad xor_sum. Because the uLan protocol is relatively slow and it is designed to be used for

real-time control, it would be bad to rely on such late collision detection. On the other side, usage of centralized arbiter or token passing behaves badly when some node dies and takes piece of bandwidth. That is why deterministic distributed arbitration scheme has been used.

The RS-485 physical link can express character of wired-and function if termination resistors and pull-up/down resistors are employed to ensure predetermined inactive state levels. This enables to define fully deterministic access method based on inactive and dominant level transition. The arbitration procedure uses eleven times slower transfer unit timing than data transfer which provides enough time for electric signal propagation and compensates worse speed of dominant to inactive state transition caused by bus capacity.

The deterministic arbitration is achieved by timing rules for media access sequence. The sequence consist of sending of break characters (11 bits of zero) with enabled line driver and waiting for specified intervals with disabled line driver and listening for breaks from other nodes. All times are taken as multiples of time for sending one character, so no additional timer is needed. The node must wait for 4 to 20 character times of no activity on line before start of the arbitration sequence. This time is calculated from difference of own address and address of last node owning line. The last owner address with B7 and B8 set is used as line release signal. To revitalize communication in case, that last winning node dies before sending of release signal, silence of more than four characters is taken as indication of such situation. All nodes sets interval 20 character times in such case. If node receives any character in this wait interval, it considers line as busy and must wait for release signal or long silence (die of previous master). This scheme lowers possibility of collision and its second benefit is cyclic priority between modulo 16 masters addresses with waiting messages with no decrease of total bandwidth.

To satisfy fully deterministic arbitration between 64 masters, arbitration sequence continues after first break character by sending another three ones in times fully determined by own address of master. Receiving of any character at time of

Bus request and release

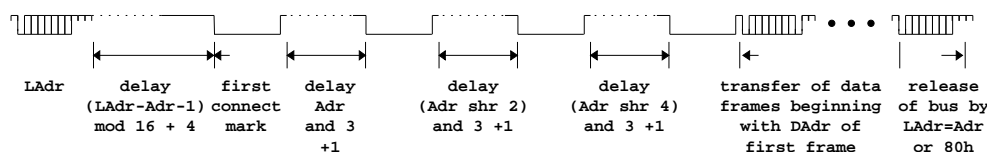


Fig. 3. uLan Access Arbitration

inter-break delay leads to lost of arbitration. Full timing specification is written in figure 3.

This solution of the access arbitration takes some time but master side implementation is significantly simpler than for Profibus protocol. The lower speed of arbitration timing enhances ten times bus length to data bit-rate ratio over CAN protocol. On the other hand, arbitration takes considerably longer time when compared with transfer time of one byte when compared to the CAN protocol. The consequence is that uLan is not well optimized for very short messages transfers of individual variables. The higher level uLan Object Interface protocol groups more properties values or description exchanges into one message and resulting communication pattern is well suited for relatively small default bit-rates used as default for uLan communication.

4. END USER APPLICATIONS

4.1 CHROMuLAN System

The CHROMuLAN project offers free software for controlling of chromatographic devices and subsequent evaluation of the acquired data. The project has been initiated and is sponsored by the PiKRON company as response for its needs to have affordable control system for its fully digital HPLC instruments. The available commercial HPLC systems software porting would mean non-affordable investments that decision has been made to build open-source system with basic functionalities together with academic users.

For incorporation of other devices (mainly detectors with analog output), an adaptor with 20-bit AD converters and TTL inputs and outputs has been developed. Due to open-source nature of the project, it is possible to create software modules (MS Windows DLLs) for communication with devices of other manufacturers.

The software is developed using Borland Delphi for Window NT/2000/XP. Porting to the Linux system is planned with use of Lazarus environment and FreePascal compiler in the future. The main developer and maintainer of the CHROMuLAN software is Jindrich Jindrich, assistant professor at the Department of Organic Chemistry, Charles University, Prague.

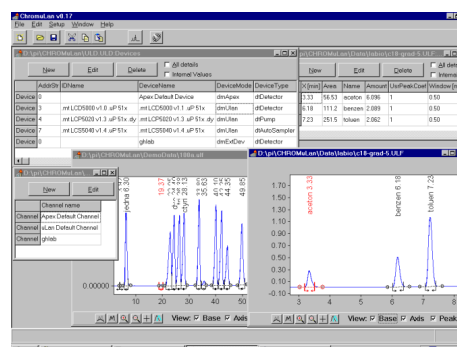


Fig. 4. CHROMuLAN User Interface

CHROMuLAN System Features

- Chromatographic assay quantitative and qualitative evaluation (retention time, area, height, half-width computation, theoretical plates) and protocols printing
- Manual and automatic peaks and baseline detection, component names assignment
- Support for multi-level calibration selectable for each component by means of composed calibration files (*.ULC). Calibration defined as set of line segments or approximated by analytical functions (linear, quadratic, etc. using of least squares method)
- Digital data acquisition through uLan communication bus or plug-in DLL modules
- Instruments control control, the time and gradient program preparation. Delivering of precompiled gradient sequences to the HPLC pump. Interpretation of more complex commands and Pascal language scripts on the PC side
- Support for sequences of samples evaluation and the autosampler control
- Number of the parallel running channels is limited only by computer speed memory, storage and communication interfaces
- All instruments setup parameters and time programs used for chromatographic assay control are archived together with acquired chromatogram data and evaluation method parameters into (*.ULF) file. This enables to fully audit even archival data and use any known good analysis as template for new one. Concept of set of templates describing all conditions coexist with ability to export and import standalone methods and instruments setup files as well.

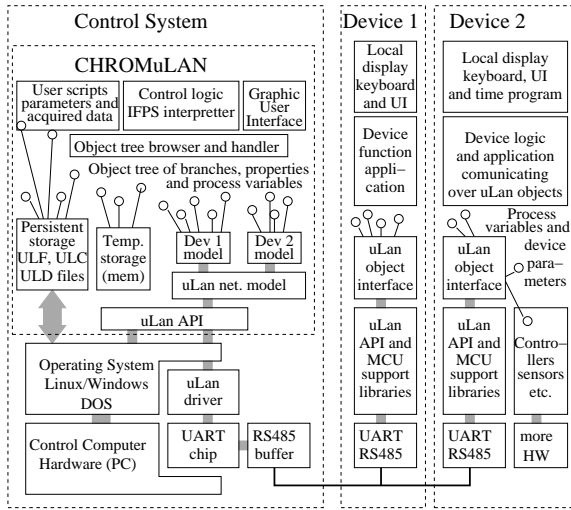


Fig. 5. CHROMuLAN System Diagram

The communication with instruments over uLan network is achieved through reading and writing device models properties. The models of the devices connected to the uLan network is build automatically from process variables and properties description exported by each instrument. The description contains property name, data type and support of read and or write functions. Even control of previously unknown device is fully supported by the CHROMuLAN systems because model of device is build during phase of device recognition. All properties of each device are listed in one node of the object tree representing all processing and acquisition data in the CHROMuLAN system. The user controls and time programs manipulates with properties by calling interpreted Pascal scripts which could be prepared or modified to fully support new device without need to touch CHROMuLAN system base. Structure of CHROMuLAN system and instruments/devices and its objects interfaces is shown in Figure 5. The more information and the CHROMuLAN project related information can be found on CHROMuLAN project pages (3).

4.2 Automatic Feeding System

Agriculture is an another application area where the uLan communication protocol is used in some distributed applications. The Czech company Agrosoft Tábor is producer of FASTOP and FASTOS systems for automatic batch distribution of feed to pigs and cows. The structure of each system consists of an control computer and units for feeding measure and livestock identification. The computer is common PC running Windows operating system and set of communication and user application programs. The communication program receives requests for feeding from the feeding units over uLan communication driver together. The animal RF identification is part of

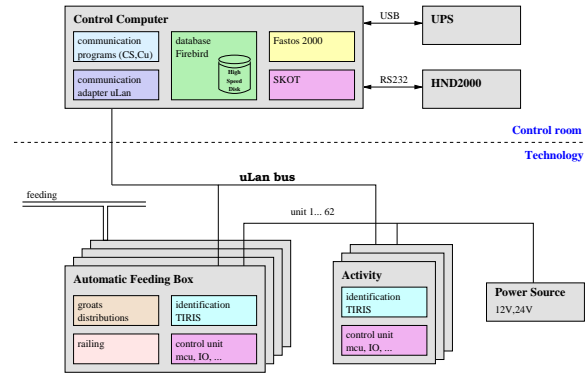


Fig. 6. FASTOP Pigs Feeding System

the request. The communication program sends replays back on the base of per animal information stored in Firebird database. Up to 62 units can be connected to one uLan bus. The system takes advantage of uLan multi-master architecture. The control system is not required to periodically poll all units. Each feeding unit contains sensor for measure of remaining feed in a scuttle. When the scuttle is empty the control unit send request directly to unit which is responsible for transport of feed. No information exchange with server is required. The structure of FASTOP system is shown in Figure 6.

4.3 Home Automation

The multi-master capability of uLan, very low cost interconnection with use of an phone line grade cable, free bus topology, non problematic interfacing to the most low cost microcontrollers and stable drivers for PC operating systems are features which speaks for spreading of uLan into other areas as well. uLan has is not intended for high speed communication or hard real-time data exchange but these features are not required for most task of home automation systems. uLan can be used for heating monitoring and control, lights switching and ring-bells in this area.

5. CONCLUSION

This paper provides basic facts about uLan protocol and documents its usability on two application which profits from its capabilities. The design and specification have been intended to be open from the beginning. The protocol software implementation for Linux, Windows and embedded microcontrollers is available under open source license on SourceForge site (2). The authors would invite others partners to use the protocol to solve their communication problems and join uLan protocol adopters group.

References

- [1] P. Píša ul.driv - ulan rs-485 communication driver, 2001.
http://cmp.felk.cvut.cz/~pisa/ulan/ul_drv.html
- [2] uLan project SourceForge pages
<http://ulan.sourceforge.net/>
- [3] CHROMuLAN project home-page
<http://www.chromulan.org/>
- [4] Jindřich, J., Píša, P., (2002). Řídicí a vyhodnocovací software pro analytickou a preparativní separaci. In.: *Pokroky v chromatografii a elektroforéze - CHIRANAL 2002*. Alga Press, Olomouc 2002