VLC Serial Communication

Edit

Last edited by **Stefan** 1 month ago

A detailed specification of the serial communication commands of ETH's VLC devices are described in the following. Configuration commands and message events are specified.

Commands

• Click to expand

Commands are always terminated by a newline character `\n`. Commands consist of one character `p`. Command parameters are provided in square brackets `a[FF]`.

Command	Description	Example	Response
`r`	resets the device	`r`	none
`p`	returns the software version string	`p`	`p[v1.0.1]`
`a / a[XX]`	returns the device address or sets the device address (in hex)	`a / a[AB]`	`a[AB] / a[AB]`
`c[grp,par,val]`	configures device parameters, echoes command if processed	`c[1,1,16]`	`c[1,1,16]`
`m[msg,dest]`	dispatches message (10 terminated) with destination (in hex), returns 1 on success, 0 on drop	`m[hello\0,AB]`	`m[P,1,AB] / m[P,0,AB]`

Configuration Commands

• Click to expand

Configuration commands are of form `c[group,parameter,value]`.

PHY grp [0]	Description	MAC grp [1]	Description	LOG grp [2]	Description
`0`	PHY preamble length	`0`	# of re-transmissions	`0`	logging level (0: none, 7: silly)
`1`	FEC threshold (default: disabled)	`1`	DIFS	`1`	logger prefix character (default: disabled)
`2`	Channel busy threshold (default: 20)	`2`	CWmin (use power of two)		
`3`	light emission (enable/disable)	`3`	CWmax (use power of two)		
		`4`	RTS threshold (default: disabled)		
		`5`	MAC address		

Message Events

• Click to expand

Message events are of form `m[X(,type,message)]`. Message events `m[T]` and `m[R,…]` are always accompanied by statistic events of the form `s[mode,type,src->dest,size(txsize),seq,cw,cwsize,dispatch,time]`, where `cw`, `cwsize` and `dispatch` are only available for data transmission events (`m[T]`)

Event	Description
`m[T]`	frame transmission completed by PHY (data frames, control frames), triggered for every transmission (also retransmissions)
`m[D]`	frame transmission done (including all retransmissions), dispatched message completely handled, new message can be dispatched
`m[R,type,message]`	frame received with type (D=DATA, A=ACK, R=RTS, C=CTS) and data payload: message

Statistics Parameters

• Click to expand

Field	Description
`mode`	R: received; T: sent
`type`	D: DATA, A: ACK, R: RTS, C: CTS-R appended for retransmission
`src`	Source address (in hex)
`dest`	Destination address (in hex)
`size`	Payload size [byte]
`txsize`	Payload size + overhead (headers, CRC, FCS) [byte]
`seq`	Sequence number
`CW`	Chosen contention window slot
`cwsize`	Current contention window size
`dispatch`	Frame dispatch time [ms] (by MAC)
`time`	Frame received/sent time [ms] (by PHY)

Practical Considerations, Good to Know

• Click to expand

Usually, when a serial connection is opened, the device is reset. Settings that were appliedare not persistent and lost after a device reset. Settings (including the device address) have to be newly applied always after the serial port was opened.

Sometimes, there is a specific problem when setting the unique device address: After opening the serial port, sometimes the device had to be reset before setting the address two times on each board while waiting five seconds between each command.

A message payload has to be terminated with a `\0`. Example: `m[hello\0,AB]`. The maximum message size is 200 characters!

All commands are terminated by a `\n`. An `\n` inside a message can be used by escaping it like `\\\n` (`\` also needs to be escaped in a string).

The Forward Error Correction (FEC) threshold must not be below `10`. FEC should not be applied to control frames.

Measurements should be performed with the logging level set to 0 (default), otherwise it will slow down the system.

Note that the LEDs should point towards each other.

Popular Libraries and Python Example

• Click to expand

Libraries

Popular libraries that help with the serial communication are:

- Matlab
- Java with RXTX or jSSC
- Python with pySerial
- Node.js with serialport

Linux

Please take note of the following feedback by a student team, related to Linux: "We first tried to set up the Arduino device running python on the Linux subsystem for Windows. This works at a first glance as pyserial recognizes the Arduino board without any issues. However, reading data from the device fails silently and made us first assume one of the Arduinos was faulty before trying it on plain Windows (wasting a lot of time). There is probably a way to make it work, but in case any future students try to attempt this it might be worth a warning."

Python Example

```
import serial
s = serial.Serial('COM4',115200,timeout=1) #opens a serial port (resets the device!)
time.sleep(2) #give the device some time to startup (2 seconds)
#write to the device's serial port
s.write(str.encode("a[AB]\n")) #set the device address to AB
time.sleep(0.1) #wait for settings to be applied
s.write(str.encode("c[1,0,5]\n")) #set number of retransmissions to 5
time.sleep(0.1) #wait for settings to be applied
s.write(str.encode("c[0,1,30]\n")) #set FEC threshold to 30 (apply FEC to packets with payload >= 30)
time.sleep(0.1) #wait for settings to be applied
s.write(str.encode("m[hello world!\0,CD]\n")) #send message to device with address CD

#read from the device's serial port (should be done in a separate program):
message = ""
while True: #while not terminated
 try:
   byte = s.read(1) #read one byte (blocks until data available or timeout reached)
   val = chr(byte[0])
   if val=='\n': #if termination character reached
     print (message) #print message
     message = "" #reset message
   else:
     message = message + val #concatenate the message
 except serial.SerialException:
   continue #on timeout try to read again
 except KeyboardInterrupt:
   sys.exit() #on ctrl-c terminate program
```


Pages 14

Search pages

Overview
Assignment 01 - Simulatio...
Assignment 02 - Data and ...
Assignment 03 - Wi-Fi IEE...
Assignment 04 - Wi-Fi Pro...
Assignment 05 - Practical ...
VLC Serial Communication
Assignment 06 - Practical ...
Assignment 07 - Mobile C...
Assignment 08 - Mobile C...
Jemula Guide
Submission Guidelines
Use of Generative AI

View all pages

Getting Help:
Course Overview
Class Material
Course @ ETH Zurich

Assignments:
01 02 03 04 05 06 07 08
Submission Guidelines
Use of Generative AI
Background Reading
Notebooks

Stefan Mangold
(stefan.mangold@inf.ethz.ch)