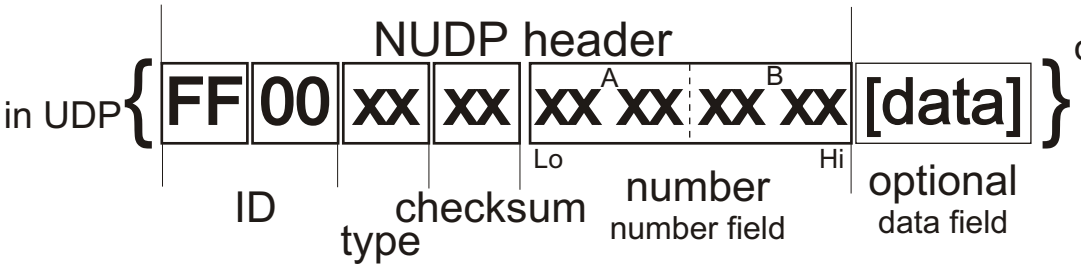
Category: Own standards for data transfers

2005.10.23

NUDP rev. 1.046 - a sensor-network protocol

It is the **internal** standard of '**Pi of the Sky**' cameras



where xx is a byte

ACK bit

can be set to 1 only by camera
0 - for top=7
1 - for all other camera responses

protocol version

(0 for this implementation)

type of packet (top)

0 - **command number / part of command** (cmd)
in *number field* (with or without parameters of the command)
(start data dumping, make a photo for example)

1 - RFU (Reserved for Future Usage)

RFU like registers in 32-bit space

2 - **set value** (request if ACK = 0),
number field consists **value number**
data field [data] consists **value to set** for camera

3 - **get value** (request if ACK = 0),
number field consists **value number**
if ACK = 1 *data field* consists **value from camera**

possible in the Future like **16-bit registers in 16-bit space**

4, 5 - according to 2, 3 where **value number** at *part A*, and **value data** at *part B* of *number field* (empty *[data]*)

6 - RAW data retransmission (request if ACK = 0),
number field consists **data packet number**
if ACK = 1 *data field* consists
data of the packet from camera

7 - RAW data always from camera (ACK = 0 always for dump!),
number field consists **data packet**
start address in 16-bit words
data field consists **data of the packet from camera**

8...15 - RFU

word and long data are in Intel format:
part A: (LO BYTE, HI BYTE)
part B: (LO, HI) (WORD)
number field in order:
(LO WORD, HI WORD)

The camera is like slave device.
data field maximum size:
default: 1024 bytes (1kB)
absolute data field size:
1500 (MTU) - 14 (Ethernet hdr) - 20 (IP hdr) - 8 (UDP hdr) - 8 (NUDP hdr) = 1450 bytes
NUDP size is always >= 8 bytes.
The checksum byte is inverted 8-bit sum of NUDP header only (~sum of first 8 bytes).
It is always calculated for all NUDP packets!

default UDP port #: 1234

8248 packets with 1024 bytes of RAW data (size X * size Y * 2 /16-bit values/)

All data from the camera are placed in NUDP data field (exception: top=5).
In response the camera modifies ACK and checksum fields in NUDP header.

The command number field consists a data which are in full compliance with clearly USB camera version.
It is needed for a backward compatibility with USB camera commands.

by NopSoft

NUDP - a sensor-network protocol

commands: (the device **has to response** on **all valid** packets with **ACK = 1**)

0x01 aa bb cc - **video processor configuration** *(see the USB cams driver documentation for details)*
aa = register of the video processor /0...7/, bb = value, cc = xx
aa = 0x00, bits of bb: 7 - ADC range (0 = 2V, 1 - 4V default); 6...3 - must be 1; 2...0 - must be 0, cc = xx
aa = 0x01, bits of bb: 7, 6, 4 - must be 1; 5...0 - must be 0, cc = xx
aa = 0x02, bb = gain RED; aa = 0x03: bb = gain BLUE (not used); aa = 0x04: bb = gain GREEN (not used); cc = xx
aa = 0x05, offset RED, bb = abs(offset), cc = sign(offset); aa = 0x06: offset BLUE (not used); aa = 0x07: offset GREEN (not used)

0x02 aa bb xx - **shutter time setting (* 10ms)**
aa = MSB, bb = LSB

0x03 xx xx xx - **start CCD readout (make a photo)**

0x04 aa bb cc - **sharpness (focusing) motor control**
aa = 0x01: bb, cc = sets number of steps (only) in left direction
aa = 0x02: bb, cc = sets number of steps (only) in right direction
aa = 0x03: bb = time between steps (* 10ms); cc = xx
aa = 0x04: motor start; bb, cc = xx
aa = 0x05: motor stop; bb, cc = xx
aa = 0x06: power on; bb, cc = xx
aa = 0x07: power off; bb, cc = xx

0x05 aa xx xx - **binning control**
aa: 1 - turn on, 2 - turn off

0x06 aa xx xx - **MPP mode control**
aa: 1 - turn on, 2 - turn off

0x07 aa bb xx - **CCD cooling control**
aa: 1 - turn on, 2 - turn off, 3 - set temperature, bb: xx or value (127 = 0°C, desired temp. + 127) for aa = 3

0x08 xx xx xx - **start RAW data dumping**
(start data transfer, here the camera must store IP address and UDP port of requester)

0x09 xx xx xx - **FPGA & USB FIFO buffers reset**

0x0A xx xx xx - **send status and temperatues request (camera returns 4 bytes in NUDP data field)**
response: 1 - CCD temperature, 2 - device status, 3 - case temp, 4 - ambient temp. *(see the USB cams driver documentation for details)*

0x0B aa xx xx - **set CCD readout speed** *(see the USB cams driver documentation for details)*
aa = coded clock frequency, bits: 2...0 horizontal, 5...3 vertical (future new command 4/4bits?)*

0x0C xx xx xx - **start CCD cleaning (CCD clear)**

0x0D aa xx xx - **gain of the pre-amplifier (LNA) control**
aa: 1 - gain x8, 2 - gain x20

0x0E aa xx xx - **lens heating control**
aa: 1 - turn on, 2 - turn off

0x0F xx xx xx - **readout the CCD with closed shutter**

0x10 aa bb cc - **ADC sampling pulse positions (or saving pulses into memory?)* - not implemented yet ***
aa = CDS1 start, bb = CDS1 stop, cc = CDS2 start, dd (in NUDP data field) = CDS2 stop (only 5 LCD used)

0x13 aa xx xx - **shutter closing mode *****
aa: 1 - shutter always opened, 2 - normal mode (open/close)

0x14 aa xx xx - **test mode control *****
aa: 1 - turn on (modulo 2^{16} test mode), 2 - turn off (normal mode), (another test modes)*

0xAA xx xx xx - **send status and temperatures request (extended - camera returns 12 bytes in NUDP data field)**
response: 1 - CCD temperature, 2 - device status, 3 - case temp, 4 - ambient temp., **
5 - LSB temp of SHT1 (ambient), 6 - MSB temp of SHT1, 7 - LSB humidity of SHT1, 8 - MSB humid of SHT1,
9 - LSB temp of SHT2 (chamber), 10 - MSB temp of SHT2, 11 - LSB humid of SHT2, 12 - MSB humid of SHT2 **

0xDD xx xx xx - **start to program ECS1** (for the USB interface only!)

0xEE 0x03 aa bb | cc - **write cc-byte to XRAM, aa, bb - address (for debug only) * / *****

0xEE 0x04 aa bb - **read a byte from XRAM, aa, bb - address (for debug only) * / *****
response: data byte from XRAM memory

0xEF xx xx xx - **send version number request (camera returns 32 bytes in NUDP data field)**
response: 1 - Year, 2 - Month, 3 - Day, 4 - Version of firmware for Cypress, 5...8 - YMDV for Altera,
9, 10 - two bytes of identification number (9 - ID from DIP-switch, 10 - 0x00), 11...32 - device name string

0xFC xx xx xx - **watchdog - command used to refresh the watchdog (Cypress watchdog timeout is 20s)**

0xFD xx xx xx - **'artifical' RESET (before Cypress will receive new firmware)** (for the USB interface only!)

where: xx - any value but 0x00 is recommended, other command numbers are reserved for the Future
*** - to discuss; ** - mistakes noticed in USB docs from 2005.09.11; *** - new commands**

by NopSoft