COMMUNICATION PROTOCOLL
DPA480P

U.A.R.T. SETTING

BAUD-RATE	= 115200
PARITY	= NONE
DATA BIT	= 8
BIT STOP	= 1

FRAME PROTOCOL

The REMOTE CONTROL must be send to the DEVICE the following frame:
TX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	..	..	Dn-1	Dn	ETX

Where:

STX	= 80H+TxByte	Start message
ID_M	= 21H	Id Device
ID_N	= 0,..,32	Id number device (0 for Usb, 1,..,32 for RS485)
CMD	= xx	Command type
DATA0,..,7,..n	= xx	Data byte
ETX	= F7H	End message

TxByte = number of byte to send

COMMAND TYPE

CMD_VERIFY DEVICE	**50H**
CMD_REQUEST_CONNECTION	**00H (lock hw GUI)**
CMD_REQUEST_DISCONNECT	**0FH (unlock hw GUI)**
CMD_UPG_GAIN	**01H**
CMD_UPG_RELATIVE_GAIN	**45H (Incr/Decr)**
CMD_UPG_DELAY	**02H**
CMD_UPG_PHASE	**03H**
CMD_UPG_XOVER	**04H**
CMD_UPG_PEQ	**05H**
CMD_BYP_EQ	**06H**
CMD_UPG_LIM	**07H**
CMD_UPG_ROUTING	**08H**
CMD_UPG_MUTE	**09H**
CMD_UPG_COMPRESSOR	**0AH**
CMD_UPG_NOISEGATE	**1AH**
CMD_ CHANGE_PRESET	**41H**
CMD_ STORE_PRESET	**28H**
CMD_READ_VERSION	**22H**
CMD_READ_X_MEM	**17H**
CMD_WRITE_X_MEM	**18H**

COMMUNICATION BETWEEN DPA480P AND REMOTE CONTROL

CMD_VERIFY DEVICE: CMD=50H

Send this command in order to understand if the device is present or not.

TX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
stx	21H	XX	50H	00H	00H	00H	00H	00H	00H	00H	00H	F7H

stx=8DH

RX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
stx	21H	XX	50H	00H	00H	00H	00H	00H	00H	00H	21H	F7H

If the device is present then the right replay is with D7 = 21H

CMD_REQUEST_CONNECTION : CMD=00H (lock hw GUI)

Send this command in order to lock the interface hw when the remote controll is connected.

TX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
stx	21H	XX	00H	00H	00H	00H	00H	00H	00H	00H	00H	F7H

stx=8DH

RX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
stx	21H	XX	00H	00H	00H	00H	00H	00H	00H	00H	26H	F7H

CMD_REQUEST_DISCONNECT: CMD=0FH
(disconnect one device connected, unlock hw GUI)

TX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
stx	21H	XX	0FH	00H	00H	00H	00H	00H	00H	00H	00H	F7H

stx=8DH

RX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
stx	21H	XX	0FH	00H	00H	00H	00H	00H	00H	00H	00H	F7H

CMD_UPG_GAIN: CMD=01H

TX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
stx	21H	XX	01H	Chn	00H	00H	00H	00H	00H	ValH	ValL	F7H

stx=8DH

Chn = 0,..,11; 0=Input1,..., 11=Output8

ValH:ValL = Gain/Volume(16bit) =0,...,300 (-18dB to +12dB step 0.1dB)

RX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX

stx	21H	XX	01H	Chn	00H	00H	00H	00H	00H	ValH	ValL	F7H

CMD_UPG_RELATIVE_GAIN: CMD=45H

TX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
stx	21H	XX	45H	Chn	Mode	00H	00H	00H	00H	00H	Step	F7H

stx=8DH
Chn = 0,..,11; 0=Input1,..., 11=Output8
Mode = 0,1; 0=decrement; 1=Increment
Step = 0,1,2,...255
The system incr/decr the current Gain (volume) with the step value (Step=1 =>0.1dB, ..., Step=10 =>1dB, ...,Step=100 =>10dB,....)
if (current Gain + step) > Max Gain the system fix the Gain to Max Gain
if (current Gain - step) < Min Gain the system fix the Gain to Min Gain

RX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
stx	21H	XX	45H	Chn	Mode	00H	00H	00H	00H	00H	Step	F7H

CMD_UPG_DELAY: CMD=02H

TX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
stx	21H	XX	02H	Chn	00H	00H	00H	00H	DlyH	DlyL	DlyF	F7H

stx=8DH
Chn = 0,..,11; 0=Input1,..., 11=Output8
if Input = DlyH:DLyL=DelayCourse (16Bit) = 0,..,420
if Output DlyH:DLyL=DelayCourse (16Bit) = 0,..,128
DlyF= DelayFine=0,..,96 (step 10.4 us)

RX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
stx	21H	XX	02H	Chn	00H	00H	00H	00H	DlyH	DlyL	DlyF	F7H

CMD_UPG_PHASE: CMD=03H

TX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
stx	21H	XX	03H	Chn	00H	00H	00H	00H	00H	00H	fPhase	F7H

stx=8DH
Chn = 0,..,11; 0=Input1,..., 11=Output8
fPhase = 0,1 (0=direct; 1=Invers (Phase 180'))

RX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX

stx	21H	XX	03H	Chn	00H	00H	00H	00H	00H	00H	fPhase	F7H

CMD_UPG_XOVER: CMD=04H

TX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
stx	21H	XX	04H	Chn	type	00H	00H	00H	FreqH	FreqL	Slope	F7H

stx=8DH
Chn = 4,..,11; 4=Output1,...,11=Output8
Type = 0,1 (0=HighPass, 1=LowPass)
FreqH+FreqL = Frequency (16bit) = 20,..,20000 (20Hz to 20KHz step 1Hz)
Slope = 0,..,12 (No cut-off, - 6 dB/oct Bw, -12dB/oct Bw, -12dB/oct LR, -12dB/oct Bs, -18dB/oct Bw, -24dB/oct Bw, -24dB/oct LR, -24dB/oct Bs, -36dB/oct Bw, -36dB/oct LR, -48dB/oct Bw, -48dB/oct LR)

RX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
stx	21H	XX	04H	Chn	type	00H	00H	00H	FreqH	FreqL	Slope	F7H

CMD_UPG_PEQ: CMD=05H

TX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
stx	21H	XX	05H	Chn	NFilt	00H	Type	FreqH	FreqL	Q	Gain	F7H

stx=8DH
Chn = 0,..,11; 0=Input1,..., 11=Output8
if Inputs => Nfilt = 0,..,29
if Outputs => Nfilt = 0,..,6
Type=0,..,16 (BELL_FILTER, HI_SHELV_1ord, HI_SHELV_2ord, HI_SHELV_Q, LO_SHELV_1ord, LO_SHELV_2ord, LO_SHELV_Q, LOW_PASS_1ord, LOW_PASS_2ord, LOW_PASS_Q, HIGH_PASS_1ord, HIGH_PASS_2ord, HIGH_PASS_Q, BAND_PASS, NOTCH_FILTER, ALL PASS lord, ALL PASS 2ord)
FreqH+FreqL = Frequency (16bit) = 20,..,20000 (20Hz to 20KHz step 1Hz)
Q=0,..,100

if Type filter EQ is BELL then

Q[100] = 0.4, 0.42, 0.45, 0.47, 0.5, 0.53, 0.56, 0.59, 0.63, 0.67,
0.71, 0.75, 0.79, 0.84, 0.89, 0.94, 1, 1.06, 1.12, 1.19,
1.26, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2, 2.1,
2.2, 2.4, 2.5, 2.7, 2.8, 3, 3.2, 3.4, 3.6, 3.8,
4, 4.2, 4.5, 4.8, 5, 5.3, 5.7, 6, 6.3, 6.7,
7.1, 7.6, 8, 8.5, 9, 9.5, 10.1, 10.7, 11.3, 12,
12.7, 13.5, 14, 15, 16, 17, 18, 19, 20, 21,
23, 24, 25, 27, 29, 30, 32, 34, 36, 38,
40, 43, 45, 48, 51, 54, 57, 60, 64, 68,

72, 76, 81, 85, 91, 96, 102, 108, 114, 121, 128
if Type filter EQ is ShelvingQ or HPQ or LPQ then
Q = (Value * 0.05) + 0.1
if Type filter EQ is BandPass or Notch then or All Pass
Q = Value + 4
Gain= Bit6,...Bit0 => 0,..,60 (-/+15dB step 0.5dB); Bit7=SByp=0,1; bypass filter (0=filter active, 1=filter bypass)

RX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
stx	21H	XX	05H	Chn	NFilt	00H	Type	FreqH	FreqL	Q	Gain	F7H

CMD_BYP_EQ: CMD=06H

TX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
stx	21H	XX	06H	Chn	00H	00H	00H	00H	00H	00H	fByp	F7H

stx=8DH
Chn = 0,..,11; 0=Input1,..., 11=Output8
fByp= 0=Not Bypass; 1=EQ Bypass

RX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
stx	21H	XX	06H	Chn	00H	00H	00H	00H	00H	00H	fByp	F7H

CMD_UPG_LIM: CMD=07H

TX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
stx	21H	XX	07H	Chn	00H	00H	00H	Byp	thr	Rel	Atk	F7H

stx=8DH
Chn = 4,..,11; 4=Output1,..., 11=Output8
Lim Byp=0,1 (0=byp Off, 1=byp On)
Lim Threshold = 0,..,150 (-10dBu to +20dBu step 0.2dBu)
Lim Release = 0,..,49
Lim Atk = 0,..,40

Rel =0.1sec to 5sec step 0.1sec

Atk = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 25, 30, 40, 50, 60, 70, 80, 90, 100, 120, 140, 160, 180, 200, 300, 400, 500, 600, 700, 800, 900 [msec]

RX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
stx	21H	XX	07H	Chn	00H	00H	00H	Byp	thr	Rel	Atk	F7H

CMD_UPG_ROUTING: CMD=08H

TX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
stx	21H	XX	08H	Chn	00H	00H	En	InA	InB	InC	InD	F7H

stx=8DH
Chn = 4,..,11; 4=Output1,..., 11=Output8
En = Enable

				InD En	InC En	InB En	InA En
0	0	0	0	0/1	0/1	0/1	0/1

InA=0,..,30; MixLevel InA to Chn (-30dB to 0dB step 1dB)
InB=0,..,30; MixLevel InB to Chn (-30dB to 0dB step 1dB)
InC=0,..,30; MixLevel InC to Chn (-30dB to 0dB step 1dB)
InD=0,..,30; MixLevel InD to Chn (-30dB to 0dB step 1dB)

RX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
stx	21H	XX	08H	Chn	00H	00H	En	InA	InB	InC	InD	F7H

CMD_UPG_MUTE: CMD=09H

TX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
stx	21H	XX	09H	Chn	00H	00H	00H	00H	00H	00H	fMute	F7H

stx=8DH
Chn = 0,..,11; 0=Input1,..., 11=Output8
fMute = 0,1 (0=Mute OFF; 1=Mute ON)

RX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
stx	21H	XX	09H	Chn	00H	00H	00H	00H	00H	00H	fMute	F7H

CMD_UPG_COMPRESSOR: CMD=0AH

TX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
stx	21H	XX	0AH	Chn	Byp	thr	rto	knee	rel	atk	m-up	F7H

stx=8DH
Chn = 0,..,11; 0=Input1,..., 11=Output8
Cmp Byp=0,1 (0=byp Off, 1=byp On)
Cmp Threshold = 0,..,150 (-10dBu to +20dBu step 0.2dBu)
Cmp Ratio = 0,..,31 (2:1,....,32:1)

Cmp Knee = 0,..,100 (from 0% (hard knee) to 100%(soft knee))
Cmp Release = 0,..,29
Cmp Atk = 0,..,29
Cmp Make-up = 0,..240 (-/+12dB step 0.1)

Rel =0.1sec to 3sec step 0.1sec

Atk = 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 25, 30, 40, 50, 60, 70, 80, 90, 100, 120, 140, 160, 180, 200 [msec]

RX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
stx	21H	XX	0AH	Chn	Byp	thr	rto	knee	rel	atk	m-up	F7H

CMD _UPG_NOISEGATE: CMD=1AH

TX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
stx	21H	XX	1AH	Chn	00H	00H	00H	Byp	Thr	Rel	Atk	F7H

stx=8DH
Chn = 0,..,3 0=Input1,..., 3=Input4
Byp= 0,1 (0=bypass Off, 1=bypass ON)
Thr = 0,..,6 (-80dBu to -50dBu step 5dBu)
Rel = 0,..,12 =[ms] "10", "25", "50", "100", "200", "300", "400", "500", "600", "700", "800", "900", "1000"
Atk = 0,..,28 = [ms] "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "15", "20", "30", "40", "50", "60", "70", "80", "90", "100","200", "300", "400", "500", "600", "700", "800", "900", "1000"

RX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
stx	21H	XX	1AH	Chn	00H	00H	00H	Byp	Thr	Rel	Atk	F7H

CMD_ CHANGE_PRESET: CMD=41H

TX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
stx	21H	XX	41H	Nprst	00H	00H	00H	00H	00H	00H	00H	F7H

stx=8DH
Nprst = 0,..,23 = preset number (Preset1,...,Preset24)

RX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
stx	21H	XX	41H	Nprst	00H	00H	00H	00H	00H	00H	00H	F7H

If Val = Nprst then preset loaded
If Val = FFH then preset not loaded

CMD_ STORE_PRESET: CMD=28H
store the current configuration to memory preset position

TX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
stx	21H	XX	28H	Nprst	00H	00H	00H	00H	00H	00H	00H	F7H

stx=8DH
Position = 0,..,23 = preset number (Preset1,...,Preset24)

RX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
stx	21H	XX	28H	Nprst	00H	00H	00H	00H	00H	00H	00H	F7H

CMD_READ_VERSION: CMD=22H

TX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
stx	21H	XX	22H	00H	00H	00H	00H	00H	00H	00H	00H	F7H

stx=8DH

RX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
stx	21H	XX	22H	a	b	c	00H	00H	00H	00H	00H	F7H

Version = a.b.c

CMD_READ_X_MEM: CMD=17H

TX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX
8DH	21H	XX	17H	adr24	adr16	adr8	Len	00H	00H	00H	00H	F7H

stx=8DH
Adr24+Adr16+Adr8 = Start Address mem (24bit)
Len = 8,16,32 = num Byte to read

RX:

RxByte = 5 + Len

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	..	..	..	Dn	ETX
8DH	21H	XX	17H	d0	d1	d2	d3	..	..	..	dlen-1	F7H

CMD_WRITE_X_MEM: CMD=18H

TX:

TxByte = Len → Stx=80H + Len + 8 = 88H + Len

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	..	..	Dn	ETX
stx	21H	XX	18H	adr24	adr16	adr8	Len	d0	..	..	dlen-1	F7H

Adr24+Adr16+Adr8 = Start Address mem (24bit)
Len Max= 32 num Byte to write
Data start from D4=d0 to Dn=Dlen-1

RX:

STX	ID_M	ID_N	CMD	D0	D1	D2	D3	D4	D5	D6	D7	ETX

stx	21H	XX	18H	adr24	adr16	adr8	Len	d0	d1	d2	d3	F7H

PRESET MAP see document “DPA480P MemoryMap.ods”