

Communication Protocol The controller RSPort V.1.27

1. Introduction

Communication between HOST (ex. PC) and the amplifier the controller uses asynchronous transmission protocol. Setup of the communication port should look like the example:

Bits per second	19.200
Data bits	8
Parity	NONE
Stop bit	1
Flow controls	NONE

Host and the controller exchange information in frame form. The polling protocol is used for communication. The host is working as a master and the controller as a slave. The host has to initiate the transmission sending command frame to the controller. Then the controller executes the command and sends a response to the host. The controller never initiates a transaction, it only sends one frame for a command from the host. If the host sends an incorrect frame, the controller will send a frame REJ.

2. General frames format

Frame consists of five fields:

<i>Field</i>	<i>Length</i>	<i>Function</i>
HEAD	1 byte	Header of frame=96H
LEN	1 byte	length of frame: number of bytes in fields:<CTRL,DATA,CRC> LEN=<2,14>
CTRL	1 byte	command code
DATA	<0 , 12> bytes	command parameters
CRC	1 byte	Cyclic Redundancy Check, the generator polynomial is $x^8+x^5+x^4+1$. CRC is counting from fields = HEAD, LEN, CTRL, DATA, CRC.

The frame is variable in length, number of bytes in a frame are between 4 and 14.

3. Used frames

Figure 3.1

Host to The controller			
Mnemonic	CTRL	LEN	Description
SetLIMITS	2	10	Set Limits.
SetPAGC	3	4	Set power level for AGC mode.
SetPMGC	4	4	Set power level for MGC mode.
SetFREQ	5	6	Set frequency.
SetSKEY	7	3	Set soft key.
SetBurstPar	8	7	Set burst parameters.
SetSweepPar	9	13	Set sweep parameters.
GetLIMITS	18	2	Get Limits.
GetPAGC	19	2	Get power level for AGC mode.
GetPMGC	20	2	Get power level for MGC mode.
GetFREQ	21	2	Get frequency.
GetSKEY	23	2	Get soft key.
GetBurstPar	24	2	Get burst parameters.
GetSweepPar	25	2	Get sweep parameters.
GetSVER	29	2	Get the controller Soft/Hard version.
GetMEAS	30	2	Get measurements.
GetSTA	31	2	Get status.
The controller to Host			
ShowLIMITS	2	10	Show Limits.
ShowPAGC	3	4	Show power level for AGC mode.
ShowPMGC	4	4	Show power level for MGC mode.
ShowFREQ	5	6	Show frequency.
ShowSKEY	7	3	Show soft key.
ShowBurstPar	8	7	Show burst parameters.
ShowSweepPar	9	13	Show sweep parameters.
ShowSVER	13	8	Show the controller Soft/Hard version.
ShowMEAS	14	5	Show measurements
ShowSTA	15	10	Show status.
REJ	42	2	Unknown frame.

4. Frames

4.1. Frames from the host to the controller

4.1.1. SetLIMITS - Set limits.

HEAD	LEN	CTRL	DATA								CRC
0x96	10	2	FPLH	FPLL	RPLH	RPLL	nu	nu	nu	nu	xx

In response the controller sends **ShowLIMITS** frame.

FPLH- HighByte,

FPLL- LowByte,

<FPLH, FPLL>=ForwardPowerLimit = two bytes set for ForwardPowerLimit. Scale is dW, example ForwardPowerLimit=1=0.1W

RPLH- HighByte,

RPLL- LowByte,

<RPLH, RPLL>=ReversePowerLimit = two bytes set for ReversePowerLimit. Scale is dW, example ReversePowerLimit=1=0.1W

nu – not used.

4.1.2. SetPAGC - Set power level for AGC mode.

HEAD	LEN	CTRL	DATA		CRC
0x96	4	3	AGCPoH	AGCPoL	xx

In response the controller sends **ShowPAGC** frame.

AGCPoH - HighByte

AGCPoL - LowByte

<AGCPoH, AGCPoL> = AGCPowerLevel= two bytes set power level for AGC mode. Scale is dW, example AGCPowerLevel=1=0.1 W

4.1.3. SetPMGC - Set power level for MGC mode.

HEAD	LEN	CTRL	DATA		CRC
0x96	4	4	MGCPoH	MGCPoL	xx

In response the controller sends **ShowPMGC** frame.

MGCPoH - HighByte

MGCPoL - LowByte

<MGCPoH, MGCPoL> = MGCPowerLevel=two bytes set power level for MGC mode. Scale is 0.1%, example MGCPowerLevel=1=0.1 %

4.1.4. SetFREQ - Set internal source frequency.

HEAD	LEN	CTRL	DATA				CRC
0x96	6	5	FreqH	FreqL	FreqHzH	FreqHzL	xx

In response the controller sends **ShowFREQ** frame.

FreqH- HighByte,

FreqL- LowByte,

FreqHzH- HighByte,

FreqHzL- LowByte,

<FreqH, FreqL> =Freq =

two bytes Frequency setting.

Scale is 1kHz, meaning Freq=1=1 kHz

<FreqHzH, FreqHzL> =FreqHz=

two bytes Frequency setting.

Scale is 1Hz increments

meaning FreqHz=1=1 Hz

Current Frequency setting is equal:

Frequency=Freq*1000+FreqHz [Hz].

4.1.5. SetSKEY - Set softkey.

HEAD	LEN	CTRL	DATA				CRC
0x96	3	7	SoftKey				xx

In response the controller sends **ShowSKEY** frame.

Tab.3.4. SoftKey bits.

SoftKey		
Bit	Name	Comments
7	SoftOn	=1 Host takes over the keyboard of the controller. =0 The controller takes over the keyboard of the controller. If you want control the keyboard by the host and the controller, you will send first command SetSKEY with SoftOn=1, then wait for response. Afterwards send SetSKEY with SoftOn=0.
6	Nu	Reserved
5	Nu	Reserved
4	Nu	Reserved
3	Key1	=1 Key1 is on, =0 Key1 is off.
2	Key0	=1 Key0 is on, =0 Key0 is off.
1	Key2	=1 Key2 is on, =0 Key2 is off.
0	Key3	=1 Key3 is on, =0 Key3 is off.

4.1.6. SetBurstPar - Set burst mode parameters

HEAD	LEN	CTRL	DATA				CRC	
0x96	7	8	SCode	BRepTH	BrepTL	BOnTH	BOnTL	xx

In response the controller sends **ShowBurstPar** frame.

SCode = 0 Burst mode OFF request,
 = 1 Burst mode ON request,
 = 2 Change burst parameters without changing burst on/off.
 BRepTH- HighByte,
 BRepTL- LowByte,
 < BRepTH ,BRepTH >= BRep Repetition period burst cycle. Scale is 1ms.
 BRep range=<1ms, 50ms>.
 BOnTH- HighByte,
 BOnTL- LowByte,
 < BOnTH ,BOnTH > = BOn Time of Power nn in burst cycle. Scale is 1us.
 BOn range=<1us, 500us>

4.1.7. SetSweepPar - Set sweep mode parameters

HEAD	LEN	CTRL	DATA					CRC	
0x96	13	9	SCode	SStr	SStp	SCycH	SStrHz	SStpHz	xx

In response the controller sends **ShowSweepPar** frame.

SCode = 0 Sweep mode OFF request,
 = 1 Sweep mode ON request,
 = 2 Change sweep parameters without changing sweep mode on/off.

BRepTH- HighByte,
 BRepTL- LowByte,
 < BRepTH ,BRepTH >= BRep Repetition period burst cycle. Scale is 1ms.
 BRep range=<1ms, 50ms>.

SStr= [Word=HighByte, LowByte] Start frequency in the Sweep mode. Scale is kHz,
 SStp=[Word=HighByte, LowByte] Step frequency in the Sweep mode. Scale is kHz,
 Scyc= [Word=HighByte, LowByte] Number of steps in full sweep cycle.

SStrHz=[Word=HighByte, LowByte] Two bytes SStr offset. Scale in Hz.
 SStrHz range=<0,999>Hz
 SStpHz=[Word=HighByte, LowByte] Two bytes SStp offset. Scale in Hz.
 SStpHz range=<0,999>Hz

Current start frequency setting is equal:
 StartFrequency=SStr*1000+SStrHz [Hz]
 Current frequency step setting is equal:
 StepFrequency=SStp*1000+SStpHz [Hz]

4.1.8. GetLimits – Get limits

HEAD	LEN	CTRL	CRC
0x96	2	18	Xx

In response the controller sends **ShowLIMITS** frame.

4.1.9. GetPAGC – Get power level for AGC mode.

HEAD	LEN	CTRL	CRC
0x96	2	19	Xx

In response the controller sends **ShowPAGC** frame.

4.1.10. GetPMGC – Get power level for MGC mode.

HEAD	LEN	CTRL	CRC
0x96	2	20	Xx

In response the controller sends **ShowPMGC** frame.

4.1.11. GetFREQ – Get internal source frequency.

HEAD	LEN	CTRL	CRC
0x96	2	21	Xx

In response the controller sends **ShowFREQ** frame.

4.1.12. GetSKEY – Get soft key

HEAD	LEN	CTRL	CRC
0x96	2	23	Xx

In response the controller sends **ShowSKEY** frame.

4.1.13. GetBurstPar – Get Burst Parameters

HEAD	LEN	CTRL	CRC
0x96	2	24	Xx

In response the controller sends **ShowBurstPar** frame.

4.1.14. GetSweepPar – Get Sweep Parameters

HEAD	LEN	CTRL	CRC
0x96	2	25	Xx

In response the controller sends **ShowSweepPar** frame.

4.1.15. GetSVER – Get the controller software and hardware version

HEAD	LEN	CTRL	CRC
0x96	2	29	Xx

In response the controller sends **ShowSVER** frame.

4.1.16. GetMEAS – Get measurements

HEAD	LEN	CTRL	CRC
0x96	2	30	Xx

In response the controller sends **ShowMEAS** frame.

4.1.17. GetSTA – Get status

HEAD	LEN	CTRL	CRC
0x96	2	31	Xx

In response the controller sends **ShowSTA** frame.

4.2. Frames from the controller to host

4.2.1. ShowLIMITS - Show limits

HEAD	LEN	CTRL	DATA								CRC
0x96	10	2	FPLH	FPLL	RPLH	RPLL	nu	nu	nu	nu	xx

FPLH- HighByte,

FPLL- LowByte,

<FPLH, FPLL>=ForwardPowerLimit = two bytes set for ForwardPowerLimit. Scale is dW, example ForwardPowerLimit=1=0.1W

RPLH- HighByte,

RPLL- LowByte,

<RPLH, RPLL>=ReversePowerLimit = two bytes set for ReversePowerLimit. Scale is dW, example ReversePowerLimit=1=0.1W

nu – not used.

4.2.2. ShowPAGC - Show power level for AGC mode.

HEAD	LEN	CTRL	DATA		CRC
0x96	4	3	AGCPoH	AGCPoL	xx

AGCPoH - HighByte

AGCPoL - LowByte

<AGCPoH, AGCPoL> = AGCPowerLevel= two bytes set power level for AGC mode. Scale is dW, example AGCPowerLevel=1=0.1 W

4.2.3. ShowPMGC - Show power level for MGC mode.

HEAD	LEN	CTRL	DATA		CRC
0x96	4	4	MGCPoH	MGCPoL	xx

MGCPoH - HighByte

MGCPoL - LowByte

<MGCPoH, MGCPoL> = MGCPowerLevel=two bytes set power level for MGC mode. Scale is 0.1%,
it means MGCPowerLevel=1=0.1 %

4.2.4. ShowFREQ - Show internal source frequency.

HEAD	LEN	CTRL	DATA				CRC
0x96	6	5	FreqH	FreqL	FreqHzH	FreqHzL	xx

FreqH - HighByte,

FreqL - LowByte,

FreqHzH - HighByte,

FreqHzL - LowByte,

<FreqH, FreqL> =Freq = two bytes Frequency setting.

Scale is 1kHz, meaning Freq=1=1 kHz

<FreqHzH, FreqHzL> =FreqHz= two bytes Frequency setting.

Scale is 1Hz, meaning FreqHz=1=1 Hz

Current Frequency setting is equal:

Frequency=Freq*1000+FreqHz [Hz].

4.2.5. ShowSKEY - Show softkey

HEAD	LEN	CTRL	DATA	CRC
0x96	3	7	SoftKey	xx

Tab.3.4. SoftKey bits.

SoftKey		
Bit	Nazwa	znaczenie
7	SoftOn	=1 Host takes over the keyboard of the controller. =0 The controller takes over the keyboard of the controller.
6	Nu	Reserved
5	Nu	Reserved
4	Nu	Reserved
3	Key1	=1 Key1 is on, =0 Key1 is off.
2	Key0	=1 Key0 is on, =0 Key0 is off.
1	Key2	=1 Key2 is on, =0 Key2 is off.
0	Key3	=1 Key3 is on, =0 Key3 is off.

4.2.6. ShowBurstPar - Show burst mode parameters.

HEAD	LEN	CTRL	DATA	CRC
------	-----	------	------	-----

0x96	7	8	SCode	BRepTH	BrepTL	BOnTH	BOnTL	xx
------	---	---	-------	--------	--------	-------	-------	----

SCode = 0 Burst mode OFF request,
= 1 Burst mode ON request,
= 2 Change burst parameters without changing burst on/off.

BRepTH- HighByte,

BRepTL- LowByte,

< BRepTH ,BRepTH >= BRep

Repetition period burst cycle. Scale is 1ms.

BRep range=<1ms,50ms>.

BOnTH- HighByte,

BOnTL- LowByte,

< BOnTH ,BOnTH > = BOn

Time of Power nn in burst cycle. Scale is 1us.

BOn range=<1us,500us>

4.2.7. ShowSweepPar - Show sweep mode parameters.

HEAD	LEN	CTRL	DATA						CRC
0x96	13	9	SCode	SStr	SStp	SCycH	SStrHz	SStpHz	xx

SCode = 0 Sweep mode OFF request,
= 1 Sweep mode ON request,
= 2 Change sweep parameters without changing sweep mode on/off.

BRepTH- HighByte,

BRepTL- LowByte,

< BRepTH ,BRepTH >= BRep

Repetition period burst cycle. Scale is 1ms.

BRep range=<1ms,50ms>.

SStr= [Word=HighByte, LowByte]

Start frequency in the Sweep mode. Scale is kHz,

SStp=[Word=HighByte, LowByte]

Step frequency in the Sweep mode. Scale is kHz,

Scyc= [Word=HighByte, LowByte]

Number of steps in full sweep cycle.

SStrHz=[Word=HighByte, LowByte]

Two bytes SStr offset. Scale in Hz.

SStrHz range=<0,999>Hz

SStpHz=[Word=HighByte, LowByte]

Two bytes SStp offset. Scale in Hz.

SStpHz range=<0,999>Hz

Current start frequency setting is equal:

StartFrequency=SStr*1000+SStrHz [Hz]

Current frequency step setting is equal:

StepFrequency=SStp*1000+SStpHz [Hz]

4.2.8 ShowSVER - Show the controller software and hardware version

HEAD	LEN	CTRL	DATA						CRC
0x96	8	13	SNH	SNL	SVerH	SVerL	DVerH	DVerL	xx

SNH- HighByte,
 SNL- LowByte,
 <SNH, SNL>= Serial Number two bytes information about Serial Number,

SVerH- HighByte
 SVerL- LowByte,
 <SVerH, SVerL>= Software Version two bytes information about Software Version,

DVerH- HighByte,
 DVerL- LowByte,
 <DVerH, DVerL>= Device Version two bytes information about Device Version,

4.2.9. ShowMEAS - Show measurements

HEAD	LEN	CTRL	DATA								CRC
0x96	10	14	FPH	FPL	RPH	RPL	NU	NU	NU	nu	xx

FPH- HighByte,
 FPL- LowByte,
 <FPH, FPL>= Forward Power current measured ForwardPower. Scale is dW,
 it means ForwardPower=1=0.1 W

RPH- HighByte,
 RPL- LowByte,
 <RPH, RPL>= ReversePower current measured ReversePower. Scale is dW,
 it means ForwardPower=1=0.1 W

nu – not used ,

4.2.10. ShowSTA - Show current the controller state.

HEAD	LEN	CTRL	DATA			CRC
0x96	5	15	MainState	State	KeyState	xx

Bytes:

MainState	
Value	Description
0	Initialization.
1	The controller is in safe loop.
2	The controller is waiting for RFPowerOn Request in LocalMode
3	The controller is waiting for confirm of RFPowerOn in LocalMode
4	The controller is in main loop of the LocalMode.
5	The controller is waiting for RFPowerOn Request in RemoteMode
6	The controller is waiting for confirm of RFPowerOn in RemoteMode
7	The controller is in main loop of the RemoteMode.

State		
bit	Name	Description

7	FstRemote	=1 The controller works in RemoteMode, =0 The controller works in LocalMode,
6	Nu	Reserved
5	FStRFEError	=1 RFEError detected, =0 RFEError not detected,
4	fStSafetyLP	=1 error of the safety loop detected, =1 error of the safety loop not detected,
3	Nu	Reserved
2	fStExceedRP	=1 limit of the ReversePower detected, =0 limit of the ReversePower not detected,
1	fStExceedFP	=1 limit of the ForwardPower detected, =0 limit of the ForwardPower not detected,
0	fStExceedTmp	=1 temperature error detected, =1 temperature error not detected,

KeyState		
Bit	Name	Description
7	SoftOn	=1 Host takes over the keyboard of the controller. =0 The controller takes over the keyboard of the controller.
6	Nu	Reserved
5	Nu	Reserved
4	Nu	Reserved
3	Key1	=1 Key1 is on, =0 Key1 is off.
2	Key0	=1 Key0 is on, =0 Key0 is off.
1	Key2	=1 Key2 is on, =0 Key2 is off.
0	Key3	=1 Key3 is on, =0 Key3 is off.

4.2.11. REJ – rejected frame – incorrect command.

HEAD	LEN	CTRL	CRC
0x96	2	42	Xx

5. Examples.

5.1. CRC Calculation.

Function in C language:

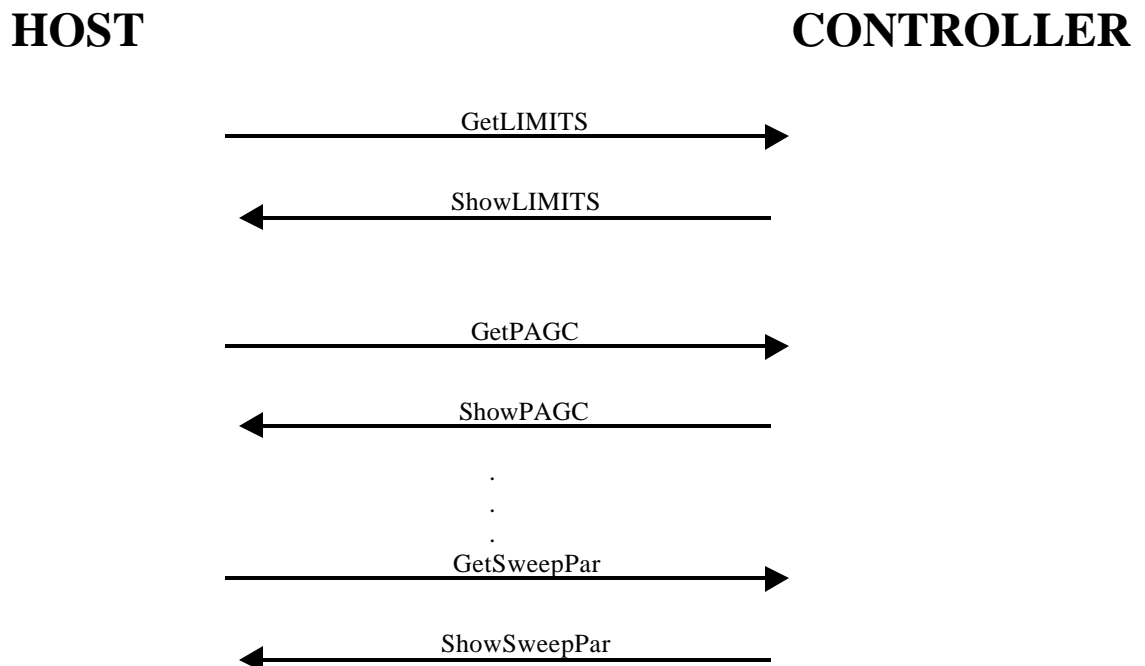
```
BYTE CalcCRC(BYTE dat, BYTE crc)
{ int i;
  for(i=0;i<8;i++)
    { if((crc ^ dat) & 0x01)
      { crc=crc ^ 0x18;
        crc=crc >> 1;
        crc=crc | 0x80;
      }
      else
      { crc=crc >> 1;
        crc=crc & 0x7f;
      }
      dat=dat >> 1;
    }
  return(crc);
}
```

In Transmitted Frame, first set variable `crc=0`, then call `CalcCRC` for fields `HEAD`, `LEN`, `CTRL` and `DATA`, and write variable `crc` in field `CRC`.

Received frame is correct if calculated CRC from fields: `HEAD`, `LEN`, `CTRL`, `DATA` are equal to CRC field from received frame.

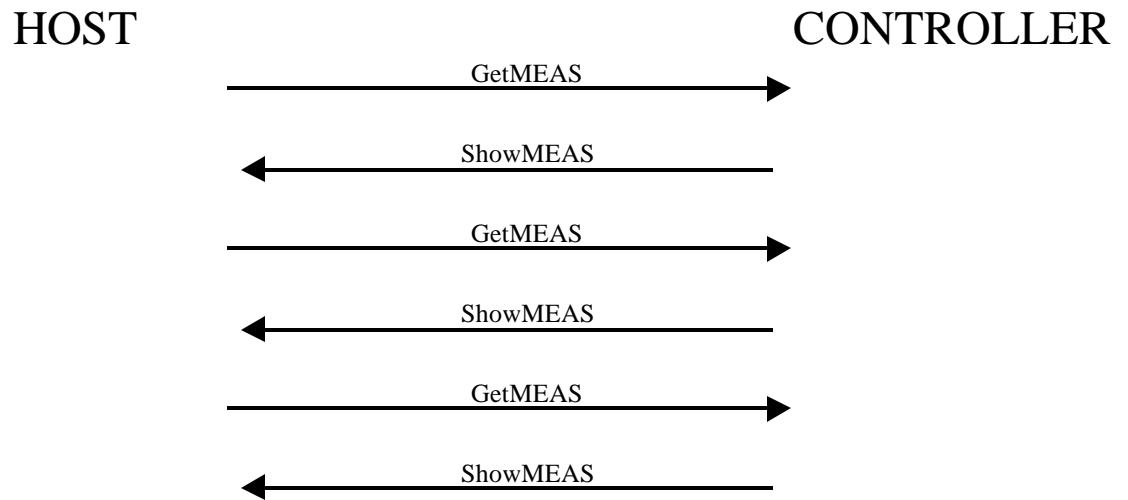
5.2.Initialization.

Exemplary transactions in the initialization stage.



5.3. Measure.

Exemplary transactions for continuous measurement:



5.4. Setting .

Exemplary transactions for setting PowerLevel for AGC mode.

