SDR-IP Interface Specification

Ver. 1.02

March 26, 2010

www.rfspace.com

This specification describes the protocol used to communicate with the SDR-IP digital receiver. A brief description of the SDR-IP architecture is provided along with a detailed description of the various command and data message formats is given.

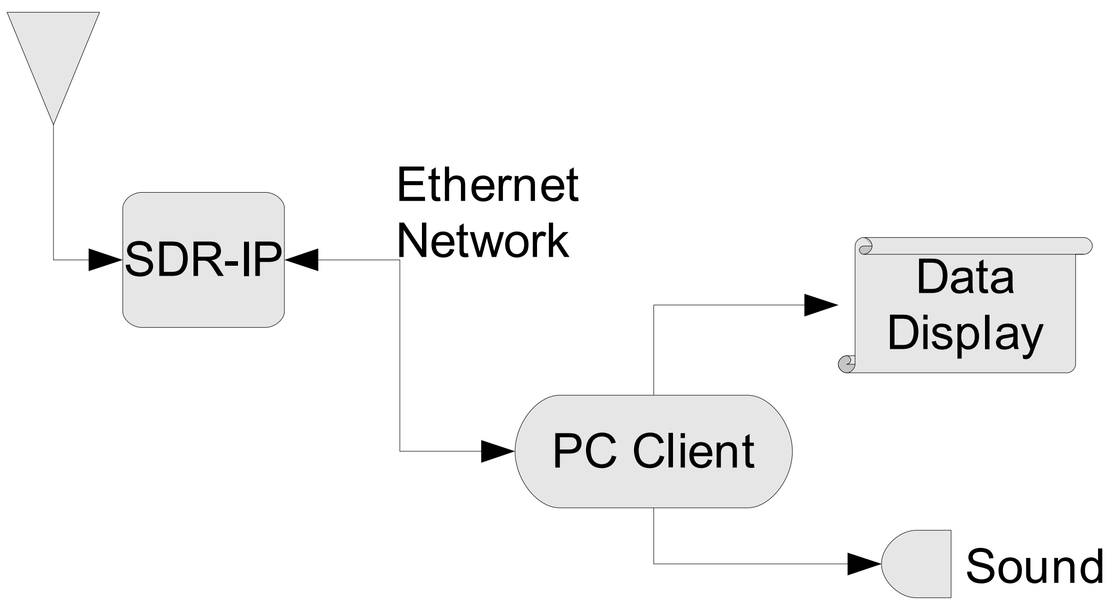
Table of Contents

1. SDR-IP Architecture

1.1. Functionality

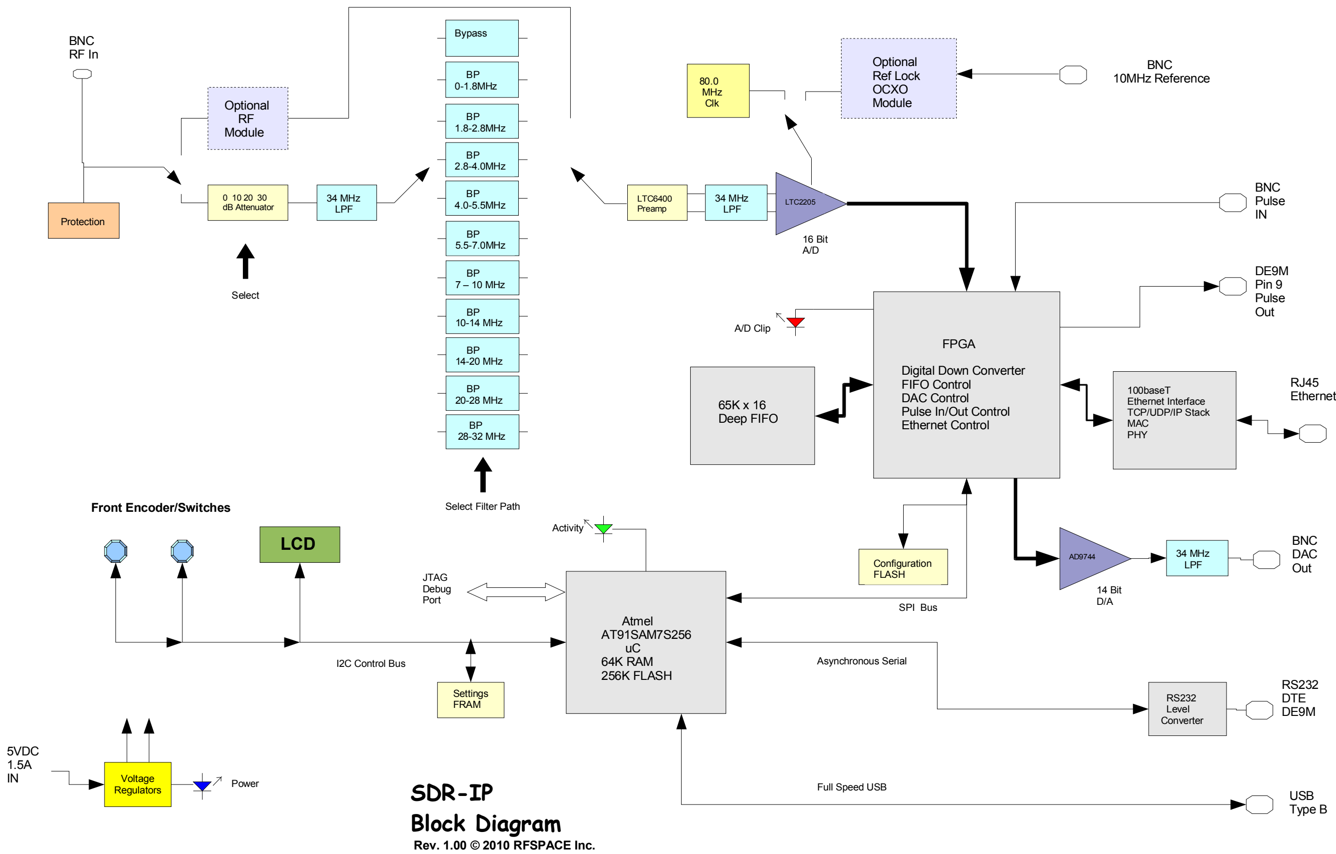
The SDR-IP is a hardware device whose basic operation is to digitize RF signals, perform various operations on the digitized data and send it back to a client application over an Ethernet network. Various modes of operations allow synchronization with external hardware control signals, analog bandpass filtering, base band down conversion ,etc.

A simple example system consisting of an SDR-IP and PC Client computer is shown below.



1.2. Block Diagram

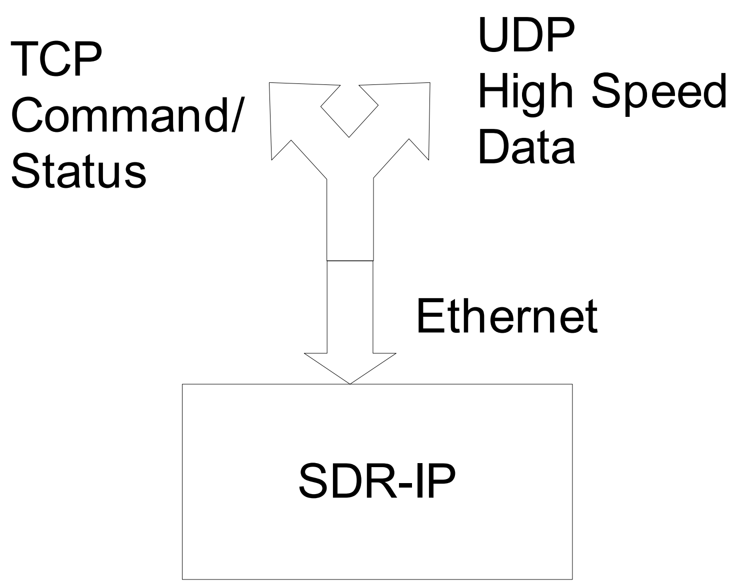
A more detailed block diagram showing the various internal function blocks.



1.3. SDR-IP Ethernet Interface

The SDR-IP runs as a TCP/IP Server allowing a single client connection. The SDR-IP primarily communicates at the "Transport Layer" or Layer 4 of the OSI Reference Model. Both TCP and UDP are used to provide command and data interfaces back to the client. A DHCP client may also be enabled that uses broadcast UDP to automatically obtain network configuration from a local DHCP server.

Even though there is only one physical Ethernet connection, one can think of the SDR-IP interface as multiple logical connections. The primary control and status messaging is done on a single TCP socket that provides guaranteed delivery of the control and status messages. The high speed digitized data is a separate one way UDP path from the SDR-IP back to the Client. This normally goes back to the same IP address as the TCP control socket but it may optionally go to a secondary IP client address on a different computer.



1.4. SDR-IP UDB Debug/Development Port

The SDR-IP has a USB port that can be used as an aid in developing/debugging interfaces to the SDR-IP. It instantiates as a simple serial COM port and then will output the SDR-IP control/status message traffic in ASCII text to any terminal program.

For the Windows platform, the SpectraVue software installation includes a USB driver for the SDR-IP debug port. It is installed at C:\Program Files\SpectraVue\SDRIPUSBdriver by default.

Other platforms should recognize the COM port automatically.

Here is some example output of a typical session. The arrow at the far left indicates the direction the message is coming from.

```
Connected0
 > Length=4    [4] [20] [1] [0] CItem Request Msg 1   CI_GENERAL_INTERFACE_NAME = SDR-IP
 < Length=11    [B] [0] [1] [0] [53] [44] [52] [2D] [49] [50] [0]
 > Length=4    [4] [20] [2] [0] CItem Request Msg 2   CI_GENERAL_INTERFACE_SERIALNUM = SD000006
 < Length=13    [D] [0] [2] [0] [53] [44] [30] [30] [30] [30] [30] [36] [0]
 > Length=9    [9] [0] [B8] [0] [0] [20] [A1] [7] [0] CI_RX_OUT_SAMPLE_RATE  500000

 < Length=9    [9] [0] [B8] [0] [0] [20] [A1] [7] [0]
 > Length=6    [6] [0] [44] [0] [0] [0] CI_RX_RF_FILTER  0

 < Length=6    [6] [0] [44] [0] [0] [0]
 > Length=6    [6] [0] [8A] [0] [0] [3] CI_RX_AD_MODES  3

 < Length=6    [6] [0] [8A] [0] [0] [3]
 > Length=6    [6] [0] [2A] [1] [0] [0] CI_RX_DA_MODES  0

 < Length=6    [6] [0] [2A] [1] [0] [0]
 > Length=6    [6] [0] [B6] [0] [0] [3] CI_RX_PULSEOUT_MODE  3

 < Length=6    [6] [0] [B6] [0] [0] [3]
 > Length=6    [6] [0] [38] [0] [0] [0] CI_RX_RF_GAIN  0

 < Length=6    [6] [0] [38] [0] [0] [0]
 > Length=10    [A] [0] [20] [0] [0] [C0] [17] [53] [2] [0] CI_RX_FREQUENCY NCO 39000000
```

2. Basic Protocol Concepts

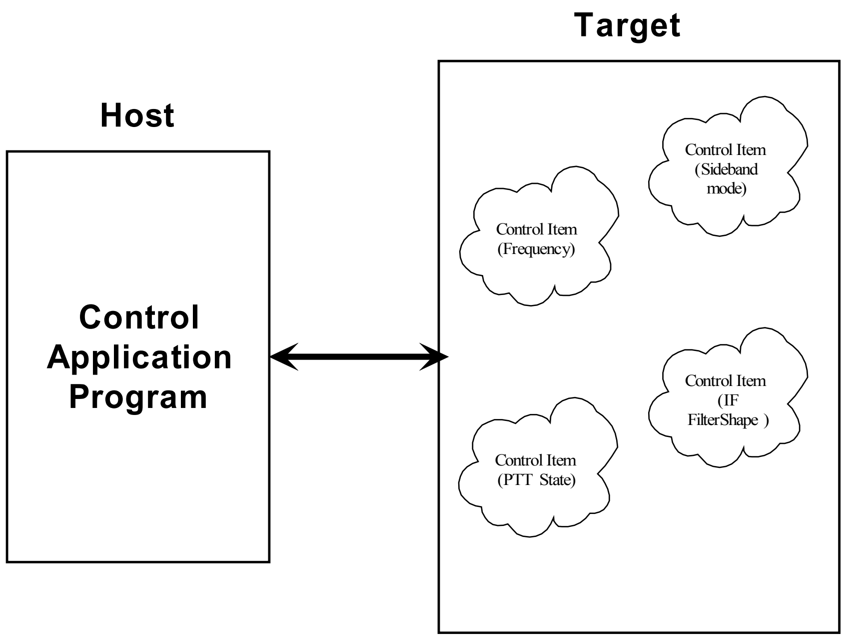
2.1. Definitions used in this specification

In the case of the SDR-IP, the host is the PC and the Target device is the SDR-IP hardware.

Host == The main initiator of communications. Typically would be a PC or other computer system such as a custom user interface controller.

Target == The device that is to be controlled or monitored by the Host in this case the SDR-IP.

Control Item == The value, setting, or state of the target that is to be controlled or monitored by the Host. For example Frequency, Sample Rate, Attenuation, Receiver state, etc.

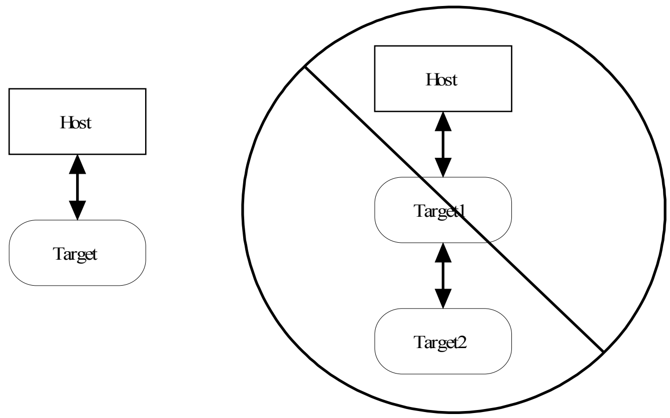


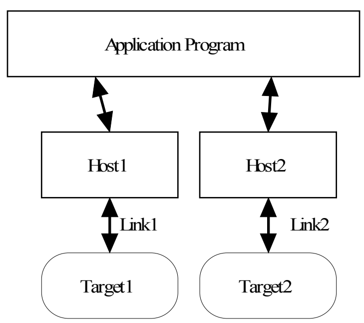
Data Item == Digital data associated with the received signal or other digital data such as firmware update.

Message Block == A contiguous block of bytes comprising a single Control Item or Data Item transfer from target to host or host to target.

2.2. Host and Target Examples

To simplify the protocol, the link can only comprise one host and one target. The Host is the only one that can set or request Control items. This means a Target device cannot connect to another target device or daisy chain to other targets. The Host can control multiple Targets by utilizing multiple links such as USB endpoints or multiple TCP/IP Sockets.





The protocol allows a Target to send unsolicited Control Item messages to the Host. This is desirable for updating the Host when a something changes in the Target without the need for polling by the Host. An example would be when a user changes the frequency of the radio using the radio's frequency knob. The target can send the updated frequency as it occurs without requiring the Host to ask for it.

Message blocks contain the block length in the message header. This is useful to aid in decoding messages as well as being able to support variable length Control Items. For example, a Control Item containing the text string for the Target's manufacturer and model number can be different lengths.

Target devices are not required to implement all the functionality of the protocol. Any unimplemented message will return a NAK response.

The Data Item message blocks allow various raw data blocks to be sent and received along with the Control Items over the same physical link or separate physical links. The Header type allows up to 4 logical channels of data to be specified in each direction. This permits sending digitized audio, digitized I/Q IF data, etc. to and from a target over the same physical connection.

Note that there is no synchronization or error handling mechanism in this protocol. This layer of protocol assumes that the block synchronization and error handling is done at a lower level. This is a reasonable assumption since Ethernet, USB, IEEE 1394, and most other modern physical links provide error recovery.

3. General Message Format

The basic message structure starts with a 16 bit header that contains the length of the block in bytes and also a 3 bit type field. If the message is a Control Item, then a 16 bit Control Item code follows the header and contains the code describing the object of the message block. This is followed by an optional number of parameter or data bytes associated with this message. The byte order for all fields greater than 8 bits is "Little Endian" or least significant byte first.

Control Item Message block format:

16 bit Header(lsb msb)	16 bit Control Item(lsb msb)	Parameter Bytes

Data Item Message block format:

16 bit Header(lsb msb)	N-Data Bytes

The 16 bit header is defined as follows:

8 bit Length lsb	3 bit type	5 bit Length msb

The 13 bit Length parameter value is the total number of bytes in the message including this header. The range of the message Length is 0 to 8191 bytes.

A special case for Data Items is that a message length of Zero is used to specify an actual message length of 8194 bytes(8192 data bytes + 2 header bytes). This allows data blocks of a power of 2 to be used which is useful in dealing with FFT data.

The message type field is used by the receiving side to determine how to process this message block. It has a different meaning depending upon whether the message is from the Host or Target.

3 bit Msg Type field	Message Source	Message Type
000	Host	Set Control Item
001	Host	Request Current Control Item
010	Host	Request Control Item Range
011	Host	Data Item ACK from Host to Target
100	Host	Host Data Item 0
101	Host	Host Data Item 1
110	Host	Host Data Item 2
111	Host	Host Data Item 3
000	Target	Response to Set or Request Current Control Item
001	Target	Unsolicited Control Item
010	Target	Response to Request Control Item Range
011	Target	Data Item ACK from Target to Host
100	Target	Target Data Item 0
101	Target	Target Data Item 1
110	Target	Target Data Item 2
111	Target	Target Data Item 3

3.1. Detailed Description of the Message Block Types and Their Purpose

3.1.1 Set Control Item

This Message type is sent from the Host to the Target requesting that the Target change the specified Control Item to the new value supplied in this message. A request to change to a new frequency would be an example of this type of message. The Target must respond to this message either with a NAK or it returns the response message(just a copy of the received Set message).

3.1.2 Request Control Item

This Message type is sent from the Host to the Target requesting that the Target respond with its current state or value of the specified Control Item of this message. A request to get the current S-meter reading would be an example of this message type. The Target must respond to this message either with a NAK or it returns the current requested control item value(s).

3.1.3 Request Control Item Range

This Message type is sent from the Host to the Target requesting that the Target respond with the acceptable range of values of the Control Item supplied in this message. A request for the targets frequency range(s) and step sizes would be an example of this message type. The Target must respond to this message either with a NAK or it returns the current control item range(s). (The SDR-xx does not implement this message type.)

3.1.4 Response to Set or Request Current Control Item

This Message type is sent from the Target to the Host in response to a request from the Host to either set or just return the current value of the Control Item supplied in this message. This message contains the current value of the Control Item. It is sent in response to either the "Set Control Item" or "Request Control Item" message.

3.1.5 Unsolicited Control Item

This Message type can be sent from the Target to the Host without any request from the Host. It contains the current value of the Control Item supplied in this message. This message can be sent at any time to the Host. It can be used to update the Host to any changes that have occurred in the Target Control Items. An example would be if the user changed frequency using the Targets frequency knob, then the Target could send the new Control Item value to the Host without having to wait for the Host to ask for it. There is no response back from the Host when this message is received.

3.1.6 Response to Request Control Item Range

This Message is sent from the Target to the Host in response to a "Request Control Item Range" message from the Host. It contains the allowable range and step size of the Control Item supplied in this message.(The SDR-xx does not implement this message type.)

3.1.7 Data Item Messages

Data Item message allow data messages to be allocated to different logical "channels". Different types of data blocks may be interleaved together and this mechanism allows each end to keep the data separated. For example, Data Item 2 blocks may be digitized audio from a Target receiver that needs to be processed and sent to a soundcard speaker. Data Item 3 Blocks may be spectral data from an FFT inside the Target receiver that needs to be sent to the Host applications display screen.

The current scheme allows up to four different logical channels for each data direction.

3.2. The ACK and NAK Messages and Their Purpose

A "NAK" message is a 16 bit header without a Control Item or parameters (Message length of 2) [02][00] . When the NAK message block is returned by the Target, it indicates that the specified Control Item is not supported. This allows a target to implement only the Control Items it actually needs. Any Host message requesting an unimplemented Control Item will be returned the NAK message. The Host can then exclude this Control Item from its list of Items to control or monitor.

As an example, suppose a Host requests the elevation setting from a rotor Target controller that only supports azimuth readings. The Target controller would just return the NAK header.

Implementation on the Target side is easily done by simply decoding only the Control Item messages that it supports and returning the NAK for all others.

On the Host side, one could initially poll the Target for all the Control Items it may use and then tag the ones that return NAK for exclusion.

A Data Item "ACK" message is a 16 bit header with a Message Type = 011b with a single parameter byte specifying the Data Item (0 to 3). The 16 bit header is a fixed value (0110 0000 0000 0011 = 0x6003). The parameter byte following the header specifies which Data Item block that is being ACK'd.

For example if the Target received a block of Data Item 2 data correctly it could send the following back to the Host:

[03][60] [02]

The ACK response messages is to provide handshaking to data item transfers. If a data item message is received correctly then an ACK response message could be sent back to the sender. This implementation is optional as one may want to stream data without error checking or only ACK periodically the data stream.

4. SDR-IP Control Item Definitions

These are all the command and control messages that are sent/received over the TCP socket connection of the SDR-IP.

All examples use hexadecimal notation within brackets [] for the individual byte values.
The "Target" referred to is the SDR-IP unit.

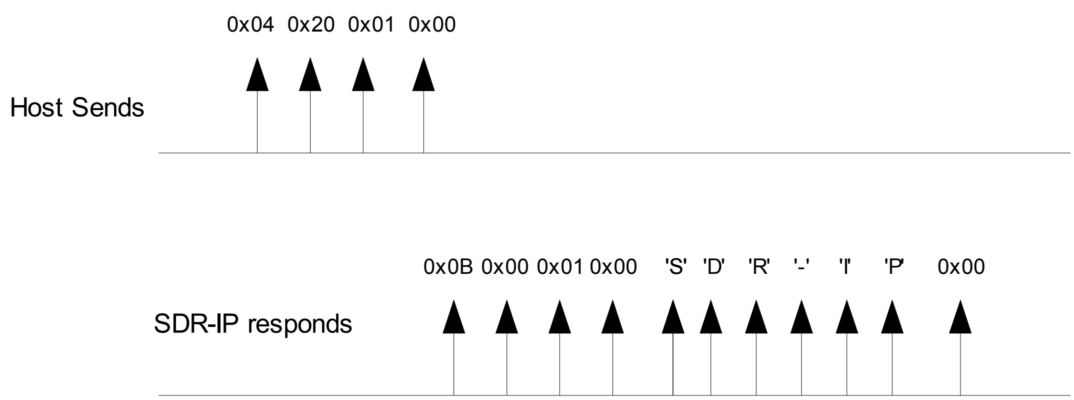
4.1. General Control Items

4.1.1 Target Name

Purpose: Returns an ASCII string describing the Target device.
Control Item Code : 0x0001
Control Item Parameter Format: The data is a NULL(zero) terminated character byte string.

Example, to request the target name, the host sends:
[04][20] [01][00]
The Target responds with "SDR-IP" :
[0B][00] [01]00] [53][44][52][2D][49][80][00]

Example Message sequence of request and response



4.1.2 Target Serial Number

Purpose: Contains an ASCII string containing the Target device serial number.
Control Item Code : 0x0002
Control Item Parameter Format: The data is a NULL(zero) terminated character byte string.

Example, to request the target serial number the host sends:
[04][20] [02][00]
The Target responds with "MT123456" or the serial number the particular device:
[0D][00] [02]00] [4D][54][31][32]33][34[35[36][00]

4.1.3 Interface Version

Purpose: Contains the version number of the Host or Targets implemented Interface. This allows the Host or Target to display or adapt to different versions of the interface.

Control Item Code: 0x0003

Control Item Parameter Format: The data is a 2 byte 16 bit unsigned variable equal to the version times 100. For example the value 123 would be version 1.23.

Example, to request the target interface version the host sends:
[04][20] [03][00]
The Target with an interface version of 5.29 responds with:
[06][00] [03]00] [11][02]

4.1.4 Hardware/Firmware Versions

Purpose: Contains the Firmware or Hardware version information of the Target.
Control Item Code: 0x0004
Control Item Parameter Format:
The first parameter is a 1 byte Firmware ID specifying which firmware or hardware version to retrieve.
ID=0 returns the SDR-IP boot code version.
ID=1 returns the SDR-IP application firmware version.
ID=2 returns the SDR-IP Hardware version.

The version data is a 2 byte 16 bit unsigned variable equal to the version times 100. For example the value 123 would be version 1.23.

Example, to request the SDR-IP firmware version host sends:
[05][20] [04][00] [01]
The Target with a SDR-IP firmware version of 5.29 responds with:
[07][00] [04]00] [01] [11][02]

Example, to request the SDR-IP boot code version host sends:
[05][20] [04][00] [00]
The Target with a SDR-IP firmware version of 5.29 responds with:
[07][00] [04]00] [00] [11][02]

4.1.5 Status/Error Code

Purpose: Contains the Error/Status code(s) of the Target. This item is used to notify the Host of any error or status change using a list of code values.

Control Item Code: 0x0005

Control Item Parameter Format: The data is a list of 1 byte unsigned variable equal to the error number associated with a particular error. There can be multiple error codes returned by the Target.
0x0B = SDR-IP Idle
0x0C = SDR-IP Busy(capturing data)
0x0D = SDR-IP Loading AD6620 parameters
0x0E = SDR-IP Boot mode Idle
0x0F = SDR-IP Boot mode busy programming
0x20 = SDR-IP A/D overload occurred
0x80 = SDR-IP Boot mode programming error

Example, host request status:
[04][20] [05][00]
The idle Target responds with
[05][00] [05][00] [0B]

If an A/D overload occurs, the SDR-IP will send an unsolicited status message back to the host.
[05][20] [05] [00] [20]

4.1.6 Product ID

Purpose: Returns the 4 byte product ID for the SDR-IP used in firmware update validation.
Control Item Code: 0x0009
Control Item Parameter Format:

A read only value returned identifying the SDR-IP.
Example, host request product ID:
[04][20] [09][00]
The Target responds with
[08][00] [09][00] [53][44][52][03]

4.2. SDR-IP Receiver Control Items

4.2.1 Receiver State

Purpose: Controls the operational state of the SDR-IP and specifies the data capture modes and formats.
Control Item Code: 0x0018
Control Item Parameter Format:

This is the main "Start/Stop" command to start or stop data capture by the SDR-IP. Several other control items need to be set first before starting the capture process such as output sample rate, packet size, etc. See the "Examples" section for typical start-up sequences.

The first parameter is a 1 byte data channel/type specifier:
Bit 7 == 1 specifies complex base band data 0 == real A/D samples
The remaining 7 bits are for future expansion and should be ignored or set to zero.
0xxx xxxx = real A/D sample data mode
1xxx xxxx = complex I/Q base band data mode

The second parameter is a 1 byte run/stop control byte defined as:
0x01 = Idle(Stop) stops the UDP port and data capture
0x02 = Run starts the UDP port SDR-IP capturing data

The third parameter is a 1 byte parameter specifying the capture mode.
Bit 7 == 1 specifies 24 bit data 0 == specifies 16 bit data
Bit [1:0] Specify the way in which the SDR-IP captures data
Bit [1:0] == 00 -Contiguously sends data as long as the SDR-IP is running.
Bit [1:0] == 01 -FIFO mode captures data into FIFO then sends data to the host then repeats.
Bit [1:0] == 11 -Hardware triggered mode where start and stop is controlled by HW trigger input
The following modes are currently defined:
0x00 = 16 bit Contiguous mode where the data is contiguously sent back to the Host.
0x80 = 24 bit Contiguous mode where the data is contiguously sent back to the Host.
0x01 = 16 bit FIFO mode where N samples of data is captured in a FIFO then sent back to the Host.
0x83 = 24 bit Hardware Triggered Pulse mode.(start/stop controlled by HW trigger input)
0x03 = 16 bit Hardware Triggered Pulse mode.(start/stop controlled by HW trigger input)

The fourth parameter is a 1 byte parameter N specifying the number of 4096 16 bit data samples to capture in the FIFO mode.

The parameter N specifies number of 4096 16 bit samples of data to be placed in the internal FIFO before being sent back to the host. It is currently only used in the 16 bit FIFO mode (Capture mode = 0x01).

Note: Some SDR-IP setup commands must be sent prior to sending this START command such as sample rate, Trigger mode, etc that are needed prior to start of capturing data.

Example: Request to start the SDR-IP capturing data in the complex I/Q base band contiguous 24 bit mode.
The host sends:
[08][00] [18][00] [80] [02] [80] [00]
The Target responds with:
[08][00] [18]00] [80] [02] [00] [00]

Example: Request to stop the SDR-IP capturing data.
The host sends:
[08][00] [18][00] [00] [01] [00] [00] //parameters 1,3, and 4 are ignored for the stop command
The Target responds with:
[08][00] [18][00] [00] [01] [00] [00]

4.2.2 Receiver Frequency

Purpose: Controls the SDR-IP NCO center frequency and SDR-IP LDC Display frequency.
Control Item Code: 0x0020
Control Item Parameter Format:
The first parameter is a 1 byte destination ID(0 or 1).
This parameter specifies whether the SDR-IP NCO frequency is specified or the LCD display frequency.
0 = SDR-IP Digital Down Converter NCO frequency (0 to 35MHz)
1 = SDR-IP LCD Display frequency (0 to 9.999 999 999 GHz)
2,3 = Future use

Followed by a 5 byte frequency value in Hz (40 bit unsigned integer LSB first)

The SDR-IP Display frequency is independent of the actual NCO frequency that it is tuned to. This allows for the use with external frequency down converters and being able to display the actual RF frequency.

Example, To set The SDR-IP NCO frequency to 14.010 MHz.
The host sends this:
[0A][00] [20][00] [00] [90][C6][D5][00][00]
The Target responds with:
[0A][00] [20][00] [00] [90][C6][D5][00][00]

Example, To get the current NCO frequency:
The host sends this:
[05][20] [20][00] [00]
The Target responds with:
[0A][00] [20][00] [00] [90][C6][D5][00] [00]

Example, To set The SDR-IP Display frequency to 7.123 456 789 GHz.
The host sends this:
[0A][00] [20][00] [01] [15][53][97][A8][01]
The Target responds with:
[0A][00] [20][00] [01] [15][53][97][A8][01]

If the user changes the Frequency knob on the SDR-IP to a frequency of 1 MHz then the following unsolicited message is sent to the Host:
[0A][20] [20][00] [01] [40][42][0F][00][00]
The Host does not respond.

4.2.3 RF Gain

Purpose: Controls the Level of RF gain(or attenuation) of the receiver.

Control Item Code: 0x0038
Control Item Parameter Format:
The first parameter is a 1 byte channel ID and is ignored.

Parameter 2 is a 1 byte signed value whose value may be (0 , -10, -20-, -30 dB).
In Hex notation these values are (0x00, 0xF6, 0xEC, 0xE2).

Example, to set the receiver RF Gain to -20 dB (-20 is 0xEC).
The host sends this:
[06][00] [38][00] [00] [EC]
The Target responds with:
[06][00] [38][00] [00] [EC]

The host sends this to request the current RF Gain setting:
[05][20] [38][00] [00]
The Target responds with:
[06][00] [38][00] [00] [EC]

4.2.4 AF Gain

Purpose: Controls the Level of AF gain LCD indicator.
Control Item Code: 0x0048
Control Item Parameter Format:
The first parameter is a 1 byte channel ID and is ignored.

Parameter 2 is a 1 byte value whose range is (0 to 16).
This value is the display level height on the LCD volume display while the SDR-IP is running.

When the volume knob on the SDR-IP is used to change the volume, this message is sent back to the Host as an unsolicited message type.

Example, to set the LCD volume display level to 10:
The host sends this:
[06][00] [48][00] [00] [0A]
The Target responds with:
[06][00] [48][00] [00] [0A]

If the user changes the Volume knob to a level of 3 then the following message is sent to the Host:
[06][20] [48][00] [00] [03]
The Host does not respond.

4.2.5 RF Filter Selection

Purpose: Controls the Analog RF Filter selection.
Control Item Code: 0x0044
Control Item Parameter Format:
The first parameter is a 1 byte channel ID and is ignored.

Parameter 2 is a 1 byte value whose range is (0 to 11).
0 = Automatically select filter based on NCO frequency.
1 = Select 0 to 1.8 MHz Filter
2 = Select 1.8 to 2.8 MHz Filter
3 = Select 2.8 to 4.0 MHz Filter
4 = Select 4.0 to 5.5 MHz Filter
5 = Select 5.5 to 7.0 MHz Filter
6 = Select 7 to 10 MHz Filter
7 = Select 10 to 14 MHz Filter

8 = Select 14 to 20 MHz Filter
9 = Select 20 to 28 MHz Filter
10 = Select 28 to 34 MHz Filter
11 = Bypass Filters

Example, to select the 5.5 to 7MHz filter:
The host sends this:
[06][00] [44][00] [00] [05]
The Target responds with:
[06][00] [44][00] [00] [05]

4.2.6 A/D Modes

Purpose: Controls various A/D Modes.
Control Item Code: 0x008A
Control Item Parameter Format:
The first parameter is a 1 byte channel ID and is ignored.

Parameter 2 is a 1 byte value whose bits are defined as follows:
Bit 0 --- 1 = Dither on 0 = Dither off
Bit 1 --- 1 = A/D gain 1.5 0 = A/D gain 1.0
Bit 2 --- 1 = Down converter input path 0 = Normal Input path

Example, to set the A/D mode to dither on and A/D gain 1.5:
The host sends this:
[06][00] [8A][00] [00] [03]
The Target responds with:
[06][00] [8A][00] [00] [03]

4.2.7 Input Sync Modes

Purpose: Controls various Hardware Input Synchronization Modes.
Control Item Code: 0x00B4
Control Item Parameter Format:
The first parameter is a 1 byte channel ID and is ignored.

Parameter 2 is a 1 byte value specifying the sync mode:
0 = No Hardware input synchronization (default)
1 = Negative Edge Start Trigger (capture starts on negative edge and runs till packet count reached)
2 = Positive Edge Start Trigger (capture starts on positive edge and runs till packet count reached)
3 = Low Level Start Trigger (capture starts on high to low transition and runs until Trigger goes high)
4 = High Level Start Trigger (capture starts on low to high transition and runs until Trigger goes low)
5 = Low Level Data Mute (data is zeroed out while Trigger level is low)
6 = High Level Data Mute (data is zeroed out while Trigger level is high)

Parameter 3,4 is a 2 byte 16bit value specifying the number of SDR-IP UDP packets to send for sync modes 0 and 1.
This 16 bit value N specifies 8*N number of UDP packets that are sent, or for example, if N = 3, then 24 UDP packets will be sent .

If the 16 bit packet count value is zero, then the SDR-IP will trigger and run continuously until stopped.

Example, to set the Input Sync mode to Neg edge mode and to send 8000 (1000*8) UDP packets:
The host sends this:
[08][00] [B4][00] [00] [01] [E8][03]
The Target responds with:
[08][00] [B4][00] [00] [01] [E8][03]

4.2.8 DDC Output Sample Rate

Purpose: Specifies the SDR-IP DDC output sample rate.
Control Item Code: 0xB8
Control Item Parameter Format:
The first parameter is a 1 byte receiver channel ID. This parameter is ignored.

The following 4 byte parameter specifies the SDR-IP DDC output sample rate in Hz.
This parameter limited to frequencies that are integer divisions by 10 of the 80MHz A/D sample rate.

This message is not needed for the FIFO mode where the direct A/D samples are captured to the SDR-IP's memory then sent back the host bypassing the Digital Down Converter(DDC).

There are limits to the range of output sample rates that are supported depending on the number of bits per sample that are specified:
The maximum sample rate supported is 2,000,000 Hz in the 16 bits/sample mode.(80MHz/40)
The maximum sample rate supported is 1,333,333 Hz in the 24 bits/sample mode.(80MHz/60)

The minimum sample rate supported is 32,000 Hz in the 16 or 24 bits/sample mode.(80MHz/250)

Example, to set the SDR-IP DDC sample rate to 500KHz (80MHz/160)
The host sends:
[09][00] [B8][00] [00] [20][A1][07][00]
The Target would reply with the following:
[09][00] [B8][00] [00] [20][A1][07][00]

4.3. SDR-IP Calibration Control Items

4.3.1 A/D Input Sample Rate Calibration

Purpose: Specifies the SDR-IP A/D input sample rate for calibration purposes.
Control Item Code: 0xB0
Control Item Parameter Format:
The first parameter is a 1 byte receiver channel ID. This parameter is ignored.

The following 4 byte parameter specifies the input source sample rate in Hz.
The SDR-IP nominally has a sample rate of 80,000,000 Hz. This command can be used to specify the actual sample rate so that the frequency can be set more accurately. This value is saved in the SDR-IP internal memory so does not need to be set unless a new calibration value is needed.
Note, this does not change the sample rate but is a way to tell the SDR-IP what its actual sample rate is so it can accurately set its NCO frequency.

Example, suppose the actual A/D sample rate is 80,000,123Hz.

To set the SDR-IP sample rate to 80,000,123Hz.
The host sends:
[09][00] [B0][00] [00] [7B][B4][C4][04]
The Target would reply with the following:
[09][00] [B0][00] [00] [7B][B4][C4][04]

4.3.2 SDR-IP Calibration Data

Purpose: Sets the DC offset value to be used by the A/D Converter.
Control Item Code: 0x00D0
Control Item Parameter Format:

The first parameter is a 2 byte signed 16 Bit value specifying the DC Offset. This value is saved in the SDR-IP internal memory so only needs to be changed when a new calibration value is needed.
This offset is used to reduce the spectral power at DC when an FFT is performed on the data.

Example, to set the A/D DC Offset to -234.
The host sends this:
[06][00] [D0][00] [16] [FF]
The Target responds with:
[06][00] [D0][00] [16] [FF]

4.4. SDR-IP Misc Control Items

4.4.1 Pulse Output Modes

Purpose: Controls various Hardware Pulse output modes (Pin 9 of DE9 Connector).
Control Item Code: 0x00B6
Control Item Parameter Format:
The first parameter is a 1 byte channel ID and is ignored.

Parameter 2 is a1 byte value specifying the pulse output mode:
0 = No Hardware output (default)
1 = Run State. High while SDR-IP is capturing, Low when Idle.
2 = Run Pulse. High when capture starts then toggles every 32768 output samples.
3 = Sample Rate. Toggles at the DDC output sample rate.

Example, to set the Output Pulse mode to Sample Rate mode:
The host sends this:
[06][00] [B6][00] [00] [03]
The Target responds with:
[06][00] [B6][00] [00] [03]

4.4.2 D/A Output Modes

Purpose: Controls various DAC output modes.
Control Item Code: 0x012A
Control Item Parameter Format:
The first parameter is a 1 byte channel ID and is ignored.

Parameter 2 is a1 byte value specifying the D/A output mode:
0 = No D/A output (default)
1 = A/D Echo. Echos the A/D data to the D/A.
2 = NCO Track. Outputs the DDC NCO sin frequency to the D/A.
3 = Noise Generator. Outputs a pseudo random noise sequence to the D/A.

Example, to set the D/A output mode to NCO tracking:
The host sends this:
[06][00] [2A][01] [00] [02]
The Target responds with:
[06][00] [2A][01] [00] [02]

4.4.3 Data Output Packet Size

Purpose: Sets the UDP data packet size for the SDR-IP.
Control Item Code: 0x00C4
Control Item Parameter Format:
The first parameter is a 1 byte value specifying the UDP Packet size(0 or 1).
Setting to a smaller packet size(MTU) may aid in routing over high speed Internet connections.

0 == Large UDP packets (1444 bytes(24bit data) or 1028 bytes(16bit data)) (default)
1 == Small UDP packets (388 bytes(24bit data) or 516 bytes(16bit data))

To set the UDP packet size to small the host sends:
[05][00] [C4][00] [01]
The Target responds with:
[05][00] [C4][00] [01]

4.4.4 Data Output UDP IP and Port Address

Purpose: Sets the UDP IP address and Port number for the SDR-IP data output.
Control Item Code: 0x00C5
Control Item Parameter Format:

The first parameter is a 4 byte Little Endian value specifying the UDP IP Address.

If this command is not sent, then the SDR-IP will use either the same IP address of the TCP Client that is connecting to it, or the UDP IP address set from the front panel of the unit.

The second parameter is a 16 bit port number in little endian format from 0 to 65535 that gets used by the UDP instead of the Port number of the TCP socket. This value is volatile and not saved on power down so should be set after connecting to the SDR-IP and before running.

To set the UDP IP address to 192.168.3.123 and port to 12345:
[0A][00] [C5][00] [7B][03][A8]C0] [39][30]
The Target responds with:
[0A][00] [C5][00] [7B][03][A8]C0] [39][30]

4.4.5 RS232 Serial port Open

Purpose: Specifies and opens the SDR-IP RS232 Serial port.
Control Item Code: 0x200
Control Item Parameter Format:

The SDR-IP serial port is opened with the following parameters. Data is received and transmitted via the Data Item 2 messages.

Parameter 1 == Port number (not used)
Parameter 2 == Data Item Number (not used, always uses data Item 2)
Parameter 3 == Number data bits (not used, always 8)
Parameter 4 == Parity 0==no parity, 1==Odd, 2 == Even
Parameter 5 == Stop bits
Parameter 6 == Flow Control(not used)
Parameter 7,8,9,10 == 32 bit bit rate value

Example, to open the SDR-IP RS232 port to 9600 bps, odd parity, 2 stop bits.
The host sends:
[0E][00] [00][02] [00] [02] [08] [01] [02] [00] [80][25][00][00]
The Target would reply with the following:
[0E][00] [00][02] [00] [02] [08] [01] [02] [00] [80][25][00][00]

4.4.6 RS232 Serial port Close

Purpose: Specifies and closes the SDR-IP RS232 Serial port.
Control Item Code: 0x201

Control Item Parameter Format:

This message closes the SDR-IP serial port.

Parameter 1 == Port number (not used)

Example, to close the SDR-IP RS232 port.
The host sends:
[05][00] [01][02] [00]
The Target would reply with the following:
[05][00] [01][02] [00]

4.5. SDR-IP Data Item Definitions

4.5.1 SDR-IP Output Data Item 0

Purpose: This is the main UDP data item message that is sent back to the host when the SDR-IP is running.
Data 0 Item Parameter Format:
All data blocks sent from the SDR-IP are sent using the UDP socket. The size and data format of the packet depends upon several settings and modes of the SDR-IP. All multi-byte data values are sent in little endian byte order.

4.5.1.1 Real 16 Bit FIFO Data

If the large MTU packet is specified:

0x04	0x84	16bit Sequence Number	1024 Data Bytes (512 16bit data samples)

If the small MTU packet is specified:

0x04	0x82	16bit Sequence Number	512 Data Bytes (256 16bit data samples)

4.5.1.2 Complex 16 Bit Data

If the large MTU packet is specified:

0x04	0x84	16bit Sequence Number	1024 Data Bytes (256 16bit I/Q data samples)

If the small MTU packet is specified:

0x04	0x82	16bit Sequence Number	512 Data Bytes (128 16bit I/Q data samples)

4.5.1.3 Complex 24 Bit Data

If the large MTU packet is specified:

0xA4	0x85	16bit Sequence Number	1440 Data Bytes (240 24bit I/Q data samples)

If the small MTU packet is specified:

0x84	0x81	16bit Sequence Number	384 Data Bytes (64 24bit I/Q data samples)

4.5.1.4 Data Format Details

The data bytes represent either two 16 bit sample values representing the I and Q data or 1 real 16 bit sample value.
The byte breakdown for the 16 bit real FIFO data mode:

R1	R1	R2	R2	R3	R3	R4	R4	………

lsb	msb	lsb	msb	lsb	msb	lsb	msb	

The byte breakdown for the 16 bit I/Q data mode:

I1 lsb	I1 msb	Q1 lsb	Q1 msb	I2 lsb	I2 msb	Q2 lsb	Q2 msb	………

The byte breakdown for the 24 bit I/Q data mode:

I1 b0	I1 b1	I1 b2	Q1 b0	Q1 b1	Q1 b2	I2 b0	I2 b1	I2 b2	Q2 b0	Q2 b1	Q2 b2	….......

The 16 bit Sequence Number starts with zero then increments by one for each packet that is sent up to 65535 (0xFFFF) then starts with the value of one and increments by one again. The Sequence Number value of zero is only sent on the very start of a capture sequence. This is useful in the Hardware Sync modes to indicate in the data stream when the trigger event occurred.

Trigger	1st pkt	2nd pkt		65535th pkt	65536th pkt	65537th pkt
0x0000	0x0001	0x0002	....	0xFFFF	0x0001	0x0002

4.6. SDR-IP Firmware Update Item Definitions

This set of items is used to update the SDR-IP firmware or FPGA code.

4.6.1 Update Mode Control

Purpose: Controls the Updating of Software or Firmware code.
Control Item Code: 0x0300
Control Item Parameter Format:
The first parameter is a 1 byte device ID (0 to 255).
This byte specifies the device or bank to program.

Parameter 2 is a 1 byte the Mode command.
0x00 == ENTER (Enter boot loader code if it is not running in it currently)
0x01 == START (Begin the update process)
0x02 == END (End update process and jump back into user code)
0x03 == ABORT (Abort update process)

Parameter 3,4,5, and 6 is a 4 byte password to protect against inadvertent programming.
The SDR-IP password is:
Parameter 3 == 0x53
Parameter 4 == 0x44
Parameter 5 == 0x52
Parameter 6 == 0x03

Example, to start the update process:
The host sends this:
[0A][00] [00][03] [00] [01] [53] [44] [52] [03]
The Target responds with:
[0A][00] [00][03] [00] [01] [53] [44] [52] [03]

4.6.2 Update Mode Parameters

Purpose: Request programming parameters from the target device.
Control Item Code: 0x0302
Control Item Parameter Format:
The first parameter is a 1 byte device ID (0 to 255).

This byte specifies the device or bank to program.

The response contains these additional bytes:
Parameter 2,3,4,5 is the Flash size in bytes (32 bit unsigned integer LSB first)

Parameter 6,7,8,9 is the Flash programming page size in bytes (32 bit unsigned integer LSB first)

Parameter 10,11,12,13 is the Flash Sector size in bytes (32 bit unsigned integer LSB first)

The host sends this to request the programming parameters from the target device 0:
[05][20] [02][03] [00]
The Target responds with: (256K byte FLASH, 256 byte page, 16384 byte sector)
[11][00] [02][03] [00] [00] [00] [04] [00] [00] [01] [00] [00] [00] [40] [00] [00]

4.6.3 Update Mode Data Item 0 Block

<u>Purpose:</u> This is the main data item message that is sent to the target containing device programming data.
<u>Data 0 Item Parameter Format:</u>

Data blocks sent to the SDR-IP from the host are a fixed size containing a 4 byte address, followed by "Flash Programming Page Size" bytes of data with the 2 byte data item 0 header.

Hdr1	Hdr2	4 byte Starting address for this data block	"Flash Programming Page Size" data Bytes

Example, to program a block of data at address 0x12345 in the SDR-IP flash. Assume the flash programming page size is 256.

The host sends this: 2 hdr bytes + 4 address bytes + 256 data bytes = 262 bytes
[06][81] [45] [23] [01] [00] [9A] [78] [56] [34] [12] ....[37]

The SDR-IP responds after programming the block into flash with:
[03][60] [00]

The host can now send another data block after receiving the ack message from the target.

5. Examples of Basic SDR-IP Operations

5.1. 24bit I/Q Continuous Capture Setup Example

The following minimal steps are required to start the SDR-IP capturing data with a 100KHz sample rate and tuned to a center frequency of 20.0MHz.

The steps 1 through 5 can be sent in any order as long as they occur prior to the start command step 6.
They also only need to be sent once before the first Start command.

Most commands can be changed while the unit is running such as frequency filters, RF gain etc. Those affecting data format or sample rate should be issued while the unit is idle.

1-Set DDC Output sample rate to 100000. (do not change after start command issued)
[09] [00] [B8] [00] [00] [A0] [86] [01] [00]

2-Set the RF Filter mode to automatic.
[06] [00] [44] [00] [00] [00]

3-Set the A/D mode to Dither and Gain of 1.5.
[06] [00] [8A] [00] [00] [03]

4-Set the NCO frequency to 20.0MHz
[0A] [00] [20] [00] [00] [00] [2D] [31] [01] [00]

5-Set the SDR-IP LCD frequency display to match NCO frequency.(optional command)
[0A] [00] [20] [00] [01] [00] [2D] [31] [01] [00]

6-Send the Start Capture command, Complex I/Q data, 24 bit contiguous mode
[08] [00] [18] [00] [81] [02] [80] [00]

UDP data packets will begin being sent back to the host after this command is received.

To stop the capture process send the Idle stop command:
[08] [00] [18] [00] [00] [1] [0] [0]