DTA-2132

High-End Satellite Receiver for PCIe

DATASHEET

February 2025

Table of Contents

1. Introduction

1.1. General Description

The DTA-2132 is a professional-grade PCIe 2.0 x4 satellite receiver card designed for flexible, high-performance reception, supporting both high bit rates and robust operation with lower code rates. It features:

- **Multi-Standard Reception** – Supports DVB-S, DVB-S2, and DVB-S2X with a wide range of modulation schemes up to 256APSK and symbol rates up to 72MBd.
- **High-Performance Demodulation** – 14-bit I/Q ADC with FPGA-based DVB-S2/S2X demodulation, minimizing CPU load.
- **DVB-S Compatibility** – Supports DVB-S demodulation via CPU-based SDR processing.
- **Wide Frequency Range** – Receives L-band signals within the 950 to 2150-MHz range.
- **Versatile Stream Handling** – Supports MPEG-2 Transport Streams, Generic Stream Encapsulation (GSE), Multiple Input Streams (MIS), VCM and ACM.
- **GSE Support** – Efficiently encapsulates IP packets in a DVB-S2 stream, reducing overhead and optimizing satellite bandwidth utilization.
- **MIS Support** – Receives multiple independent data streams within a single carrier, each with its own modulation and coding scheme.
- **BBFRAME Extraction** – Received data can be made available as BBFRAMEs output in L.3 format.
- **I/Q Sample Capture** – Enables direct reception of raw 16-bit I/Q samples for custom demodulation and signal analysis.
- **Detailed Signal Quality Metrics** – Provides MER, constellation diagrams, RF power level, SNR, BER, and other signal statistics.
- **Antenna Power & Control** – Supplies LNB power (13V/18V, with 14V/19V options for long cables) and supports DiSEqC control for dish positioning and RF distribution.
- **Extensive Software Support** – Comes with a free SDK for Windows and Linux, including drivers and the DTAPI library for application development.

With its advanced feature set, the DTA-2132 is a versatile solution for professional satellite reception, signal analysis, and monitoring applications.

2. Functional Description

2.1. Block Diagram

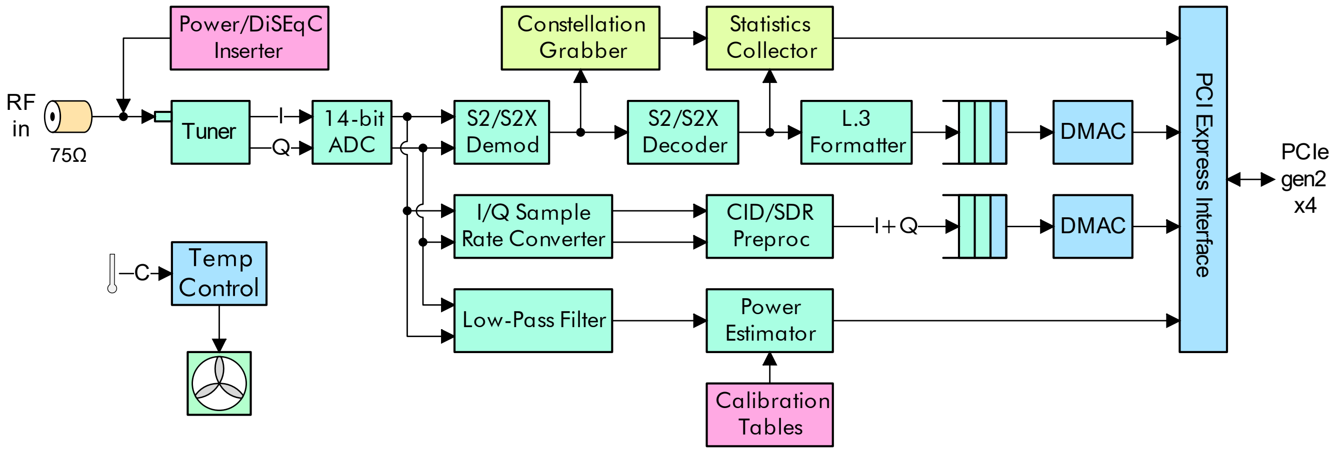

Figure 1. Functional block diagram of the DTA-2132.

2.2. Receiving Satellite Signals

The L-band signal from the LNB must be connected to the RF input port, where the DTA-2132's single tuner selects the desired signal. The tuner's center frequency is set via software.

The tuner outputs I and Q signals to a high-quality 14-bit ADC, which feeds three processing paths:

1. **DVB-S2/S2X Demodulation** – The FPGA processes the I/Q samples for DVB-S2 and DVB-S2X demodulation.
2. **I/Q Channel Output** – The raw I/Q samples can be accessed directly for custom signal analysis.
3. **Power Estimation** – The FPGA performs power measurements, providing RF power level and signal strength metrics.

Demodulation and signal measurements are performed inside the FPGA, which also reports lock status and RF quality metrics to the software. After demodulation, the data can optionally be formatted, for example, as BBFRAMEs in L.3 format, before being forwarded via the PCIe bus.

2.3. Controlling the LNB

The **DTA-2132** can be connected directly to a satellite dish, or to an RF distribution system. If connected to a dish, the card can power the LNB through the RF cable. The user can configure the LNB supply voltage and send DiSEqC commands via software to control the LNB or other RF distribution components (e.g. RF switches or motorized dishes). Refer to Sections 3.3 for detailed information.

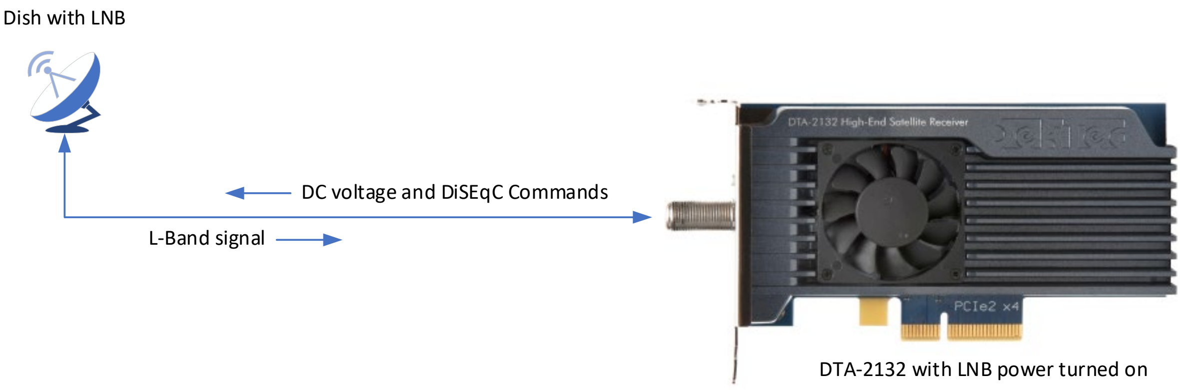

Figure 2. The DTA-2132 can receive a signal while powering/controlling the LNB over the same cable.

2.4. Software Support

The **DTA-2132** comes with a free SDK that is available for both Windows and Linux. The SDK contains a device driver and the DTAPI library that provides uniform access to any DekTec hardware. The SDK enables you to write custom applications that receive satellite signals.

The device driver implements low-level operations that require direct access to the **DTA-2132** hardware, such as initialization and coordination of DMA transfers, the handling of interrupts and reading and writing of Vital Product Data (VPD).

DekTec provides the following standard (chargeable) applications that support the **DTA-2132**:

- *StreamXpert*: Real-time stream analyzer.
- *StreamXpert Lite*: Real-time stream analyzer.
- *MuxXpert*: Real-time (re-)multiplexing.

3. Specifications

3.1. RF Input

The characteristics of the RF input are specified in the table below.

Parameter	Qualification	Min	Typ	Max	Unit
RF INPUT PORT 1					
Connector type			"F", female		
Impedance			75		Ω
Return loss	950 .. 2150MHz		-11		dB
TUNING					
Frequency range		950		2150	MHz
Sensitivity		-80		-20	dBm
STANDARDS					
DVB-S			EN 300		
DVB-S2			EN 302 307		
DVB-S2X			EN 302 307-2		
Symbol rate*	QPSK, 8-PSK, APSK, 256 APSK	0.09		72	MBd
Roll off factor			0.05, 0.10, 0.15, 0.2, 0.25, 0.35		%
Mode of operation	CCM, VCM, ACM, MIS, GS				
RF LEVEL MEASUREMENT					
Range		-80		-20	dBm
Accuracy			±3		dBm
MER MEASUREMENT					
Range		3		35	dB
Accuracy			±2		dB

* DVB-S maximum symbol rate is 27.5MBd.

3.2. Supported Modulation Parameters

The tables below specify the modulation standards, modes, code rates and symbol rates that the **DTA-2132** can properly receive. For DVB-S the minimum symbol rate is 0.09Mbd and the maximum symbol rate is 27.5Mbd. For DVB-S2 and DVB-S2X the minimum symbol rate is 0.09Mbd and the maximum symbol rate is 72Mbd.

Table 1. DVB-S2 and DVB-S2X normal frames (64800 bits).

Standard	Modulation	Code rate	Max bitrate
DVB-S	QPSK	1/2, 2/3, 3/4, 5/6, 7/8	54 Mbits
DVB-S2	QPSK	1/4,1/3, 2/5, 1/2, 2/3, 3/4, 3/5, 4/5, 5/6, 8/9, 9/10	144 Mbits
	8-PSK	2/3, 3/4, 3/5, 5/6, 8/9, 9/10	216 Mbits
	16-APSK	2/3, 3/4, 4/5, 5/6, 8/9, 9/10	288 Mbits
	32-APSK	3/4, 4/5, 5/6, 8/9, 9/10	360 Mbits
DVB-S2X	QPSK	13/45, 9/20, 11/20	144 Mbits
	8-APSK-L	5/9, 26/45	216 Mbits
	8-PSK	23/36, 25/36, 13/18	216 Mbits
	16-APSK-L	1/2, 8/15, 5/9, 3/5, 2/3	288 Mbits
	16-APSK	26/45, 3/5, 28/45, 23/36, 13/18, 7/9, 77/90	288 Mbits
	32-APSK-L	2/3	360 Mbits
	32-APSK	32/45, 11/15, 7/9	360 Mbits
	64-APSK-L	32/45	432 Mbits
	64-APSK	11/15, 7/9, 4/5, 5/6	432 Mbits
	128-APSK	3/4, 7/9	504 Mbits
	256-APSK	32/45, 3/4	576 Mbits
	256-APSK-L	29/45, 2/3, 31/45,11/15	576 Mbits

Table 2. DVB-S2 and DVB-S2X short frames (16200 bits).

Standard	Modulation	Code rate	Max bitrate
DVB-S	QPSK	1/2, 2/3, 3/4, 5/6, 7/8	54 Mbits
DVB-S2	QPSK	1/4,1/3, 2/5, 1/2, 2/3, 3/4, 3/5, 4/5, 5/6, 8/9	144 Mbits
	8-PSK	2/3, 3/4, 3/5, 5/6, 8/9	216 Mbits
	16-APSK	2/3, 3/4, 4/5, 5/6, 8/9	288 Mbits
	32-APSK	3/4, 4/5, 5/6, 8/9	360 Mbits
DVB-S2X	QPSK	11/45, 4/15, 14/45, 7/15, 8/15, 32/45	144 Mbits
	8-PSK	7/15, 8/15, 26/45, 32/45	216 Mbits
	16-APSK	7/15, 8/15, 26/45, 3/5, 32/45	288 Mbits
	32-APSK	2/3, 32/45	360 Mbits

3.3. Controlling the DISH, LNB and Upstream Equipment

The **DTA-2132** can provide both power and control to the dish, LNB and/or other upstream equipment through its RF input ports. Power for the LNB is sourced from the PCIe bus.

3.3.1. LNB Power

A DC voltage of 13V (vertical) or 18V (horizontal) is used to select polarization on an LNB. Note that the **DTA-2132** also allows 14V/19V in case there is a long cable between the receiver and the LNB.

Parameter	Qualification	Min	Typ	Max	Unit
LNB POWER (per LNB)					
Voltage	Vertical 13V	12.0	13.0	14.0	V
	Vertical 14V	13.0	14.0	15.0	V
	Horizontal 18V	17.0	18.0	19.0	V
	Horizontal 19V	18.0	19.0	20.0	V
Current	Max current drawn			600	mA
Short circuit protection	Overloading of circuit			740	mA

3.3.2. DiSEqC – Digital Satellite Equipment Control

DiSEqC (https://en.wikipedia.org/wiki/DiSEqC) is a communication protocol that allows the **DTA-2132** to control satellite equipment such as a multi-dish switch. It works by superimposing 22kHz tones on the same cable that carries the L band signal to the **DTA-2132**.

Parameter	Qualification	Min	Typ	Max	Unit
DiSEqC					
Version			v1.x*, v2.x**		
Connector type			RF, female		
Amplitude	22kHz tone	0.550	0.75	0.80	V

* DiSEqC version 1.x is a one-way protocol (from receiver to dish) that allows switching between multiple satellite sources, and, depending on the minor version, control certain other satellite receive equipment.

** DiSEqC version 2.x adds bi-directional communications.

3.4. Miscellaneous Specifications

Parameter	Qualification	Min	Typ	Max	Unit
POWER					
Supply rails used			+3.3, +12		V
Power consumption	Idle		11.8		W
	Running without LNB		12.5		W
	Running with LNBs @max		24.5		W
PCI EXPRESS BUS					
Label		PCIe2 x4			
Profile		Low profile			
MECHANICAL					
Dimensions	L x H x D card		120 x 69 x 15		mm
	L x H x D with bracket		131 x 80 x 18		mm
Weight			190		g
ENVIRONMENTAL					
Operating temperature		0		+45	°C
COMPLIANCY	In compliant PC				
CE – Emission		EN 55022:2011			
		EN 61000-3-2:2006/A1:2009			
		EN 61000-3-3:2006/A2:2010			
CE – Immunity		EN 55024:2010			
FCC – Class			B		
Safety		UL 1419, IEC60065			

4. Performance Measurements

The **DTA-2132** supports a lot of different modes, modulations and code rates. For simplicity we have sampled some of the measurements and displayed the data below. Not all measurements are presented below.

4.1. SNR and MER Measurements

The table below shows SNR and MER measurements for different modulation parameters, when a modulator with a high-quality output signal is directly connected to the RF input of the **DTA-2132**. These measurements show the maximum SNR and MER values that can be measured for each set of modulation parameters.

#	Standard	Modulation	Code rate	Roll off	FEC	Pilots	SNR (dB)	MER (dB)
A	DVB-S2	QPSK	1/2	35%	long	on or off	32.7	29.7
B		8-APSK	3/4	25%	long	on or off	32.5	29.7
C		16-APSK	5/6	35%	short	on or off	32.2	29.4
D		32-APSK	8/9	25%	long	on or off	32.2	29.4
E	DVB-S2X	8-APSK-L	5/9	5%	long	on or off	31.5	27.6
F		16-APSK-L	3/5	10%	long	on or off	30.1	29.8
G		32-APSK	5/6	15%	short	on or off	31.5	29.1
H		32-APSK-L	2/3	20%	long	on or off	31.0	28.8
I		64-APSK	11/15	25%	long	on or off	31.7	29.2
J		64-APSK-L	32/45	35%	long	on or off	31.9	29.1
K		128-APSK	7/9	5%	long	on or off	31.3	29.1
L		256-APSK	3/4	10%	Long	on or off	30.6	28.3
M		256-APSK-L	31/45	15%	Long	on or off	30.5	27.7

4.2. RF Input Port - Return Loss

The figure below shows the return loss measured at the RF input port of the DTA-2132. The maximum return loss recorded over the 900–2200 MHz frequency range is -11.10dB.

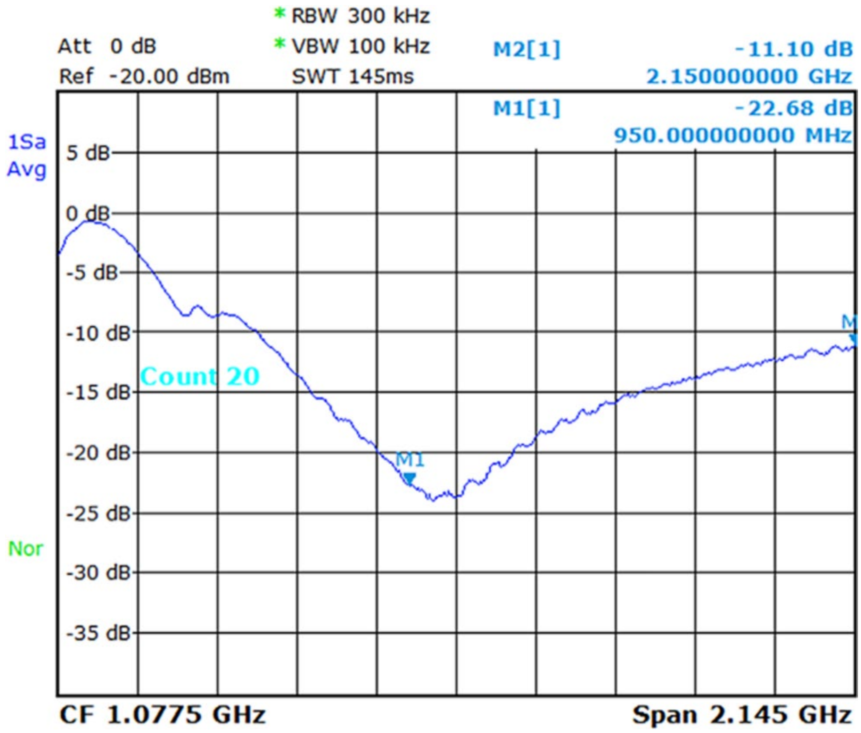

Figure 4. Return loss measurement for RF input.

5. RF Statistics

5.1. DVB-S

DVB-S Statistic	Description
Lock	True if the signal is successfully demodulated.
MER	MER of symbols.
Pre- and Post-VIT BER	Pre- and post-Viterbi bit error rate.
VIT Statistics	Viterbi error correction statistics.
Pre-RS BER	Pre-Reed-Solomon bit error rate.
RS Statistics	Reed-Solomon error correction statistics.
PER	Packet error rate after Reed-Solomon error correction.
Modulation Parameters	Detected modulation, code rate and symbol rate.
RF-level	RF power level within channel bandwidth.

5.2. DVB-S2(X)

DVB-S2(X) Statistic	Per S2(X) stream	Description
Lock – Overall		True if header lock and stream lock is true for all input streams.
Lock - PL Headers		True if physical layer headers received are successfully demodulated.
Lock – Stream	✓	True if packet payload is successfully demodulated, for a given input stream.
Lock – Percentage	✓	Percentage of locked frames, for a given input stream.
MER - PL Headers		MER of the DVB-S2 physical layer headers.
MER – Stream	✓	MER of payload symbols, for a given input stream.
MER per Baseband Frame	✓*	MER of payload symbols of the current baseband frame.
Link Margin	✓	Computed for the most complex ModCod seen in the previous time period.
Eb/N0	✓	Energy per bit to noise power spectral density ratio, based on the MER of PL Headers.
Es/N0	✓	Energy per symbol to noise power spectral density ratio, based on the MER of PL Headers.
SNR		Signal to noise ratio, based on the MER of PL Headers.
Constellation Diagram	✓	A configurable number of constellation points can be retrieved.
Pre- and Post-LDPC BER	✓	Pre- and post-LDPC bit error rate.
Pre- and Post-LDPC FER	✓	Pre- and post-LDPC frame error rate.
Pre- and Post-BCH BER	✓	Pre- and post-BCH bit error rate.
Pre- and Post-BCH FER	✓	Pre- and post-BCH frame error rate.
Modulation Parameters	✓	ModCod statistics (#frames, #occurrences per ModCod) and pilot presence in the previous time period.
RF Level		RF power level within channel bandwidth.
Spectral Inversion		True if spectral inversion is detected.
Occupied Bandwidth		Occupied bandwidth in MHz.
Roll-Off Factor		Single sample of the currently demodulated baseband header.

* Stored in L3 baseband header of each frame.

6. Data Formats

The DTA-2132 supports a wide range of data formats encapsulated within DVB-S2 and DVB-S2X frames.

Receive Mode	Format	Meaning
Transport Stream	**ST188**	188-byte transport stream packets. When the input contains 204-byte packets, the 16 trailing bytes are dropped.
	ST204	204-byte transport stream packets. When the input contains 188-byte packets, 16 zero bytes are appended.
	STMP2	188- or 204-byte transport stream packets.
	STRAW	Raw transport stream data. No notion of packets.
	STTRP	Transport stream data packaged in Transparent-Mode packets.
DVB-S2 Baseband Frame (BBFRAME)	**STL3**	DVB-S2 baseband frames with an L.3 header. Dummy frames and error frames are discarded.
	STL3FULL	DVB-S2 baseband frames with an L.3 header, including dummy frames.
	STL3ALL	DVB-S2 baseband frames with an L.3 header, including dummy frames and error frames (without payload).
Generic Stream Encapsulation (GSE)	**GSERAW**	Raw GSE-packets (ETSI TS 102 606 table 2) with an added GSE-packet header.
I/Q Samples	**IQ**	16-bit I/Q samples.

7. Developing Custom Applications with the DTA-2132 and DTAPI

Developers can create custom applications for the DTA-2132 using the free SDK provided by DekTec. The SDK, available for both Windows and Linux, includes a device driver and the DTAPI library.

The device driver implements low-level operations that require direct access to the DTA-2132 hardware, such as initiating and coordinating DMA transfers, handling interrupts, and reading and writing Vital Product Data (VPD). However, developers do not need to interact with the driver directly, as these functions are abstracted by the DTAPI library.

The standard method for receiving data from a DekTec card is described in Section 3.4 of the "DTAPI - Overview and Data Formats" document. However, additional steps are required for the DTA-2132 demodulator card.

7.1. Receiving Data

To receive data from the **DTA-2132**, you must first tune to the desired satellite channel and wait for the signal to lock. Once the lock is established, you can start receiving and processing the data.

Below is an example demonstrating how to tune to a DVB-S2 channel at 1,500 MHz and receive data:

```
// PRE-CONDITION: DtInpChannel Inp is attached to the hardware.

// Set the tuning parameters.
DtDemodPars S2DemodParams;
S2DemodParams.SetModType(DTAPI_MOD_DVBS2);
DtDemodParsDvbS2* DvbS2Params = S2DemodParams.DvbS2();
DvbS2Params->m_CodeRate = DTAPI_MOD_CR_AUTO;
DvbS2Params->m_FecFrame = DTAPI_MOD_S2_FRM_AUTO;
DvbS2Params->m_Pilots = DTAPI_MOD_S2_PILOTS_AUTO;
DvbS2Params->m_SpecInv = DTAPI_MOD_S_S2_SPECINV_AUTO;
DvbS2Params->m_SymRate = DTAPI_MOD_SYMRATE_AUTO;
int64_t FreqHz = 1'500'000'000LL;

// Start tuning.
Inp.Tune(FreqHz, &S2DemodParams);

// Wait for signal lock.
bool Locked = false;
while (!Locked)
{
    Sleep(500);
    Inp.GetStatistic(DTAPI_STAT_LOCK, Locked);
}

// Select ISI stream 0.
DtDvbS2StreamSelPars S2StreamSel;
S2StreamSel.m_ISI = 0;
Inp.SetStreamSelection(S2StreamSel);

// Set reception mode to 188-byte Ttransport-Stream packets.
Inp.SetRxMode(DTAPI_RXMODE_ST188);

// Signal the hardware to start receiving data into the receive FIFO.
Inp.SetRxControl(DTAPI_RXCTRL_RCV);

// Main loop for receiving and processing data.
while (!StopCondition())
{
    char DataBuffer[BUFSIZE];
    Inp.Read(DataBuffer, BUFSIZE);
    ProcessData(DataBuffer, BUFSIZE);
}
```

Note

- The code examples are provided for illustrative purposes only.
- In production, always validate return values and handle errors appropriately to ensure DTAPI functions operate as expected.

7.2. Powering and Controlling LNBs

The **DTA-2132** can power an LNB or other compatible equipment via its input connector. To enable power, use the `DtInpChannel::LnbEnable` method.

Below is an example demonstrating how to enable LNB power, set the LNB voltage, and send a DiSEqC switching command.

```
// PRE-CONDITION: DtInpChannel Inp is attached to the hardware.

// Ensure the input channel supports LNB control.
assert(Inp.HasCaps(DTAPI_CAP_LNB));

// Enable the LNB (maximum current: 400mA).
// The DTA-2132 ignores any optional power limiting parameter.
Inp.LnbEnable(true);

// Set LNB voltage to 18V.
Inp.LnbSetVoltage(DTAPI_LNB_18V);

// Wait briefly after enabling power to allow the DiSEqC switch to initialize.
Sleep(500);

// Send a DiSEqC command to select the second port on the DiSEqC switch.
static const uint8_t DISEQC_SWITCH_CMD_LNB2[] = {0xE0, 0x10, 0x38, 0xF4};
Inp.LnbSendDiseqcMessage(DISEQC_SWITCH_CMD_LNB2, sizeof(DISEQC_SWITCH_CMD_LNB2));
```

7.3. Retrieving the LNB Power Status

The **DTA-2132** allows retrieving LNB power status and detecting errors using the `DtInpChannel::LnbGetPowerStatus` method.

The following example demonstrates how to query and display LNB power information.

```
// PRE-CONDITION: DtInpChannel Inp is attached to the hardware.

// Ensure the input channel has the LNB capability.
assert(Inp.HasCaps(DTAPI_CAP_LNB));

// Retrieve the LNB power status.
DtLnbPowerStatus Status;
Inp.LnbGetPowerStatus(Status);

// Print the maximum LNB power.
if (Status.m_MaxPower == DtLnbPower::LNB_400MA)
    printf("Maximum LNB current is 400mA\n");
else
    // Unexpected for the DTA-2132.
    printf("Maximum LNB current is 200mA\n");

// Check and print any error flags.
if (Status.m_ErrorFlags & DTAPI_LNB_POWER_OVL)
    printf("LNB overload\n");
if (Status.m_ErrorFlags & DTAPI_LNB_THERM_OHT)
    printf("LNB overheated\n");
if (Status.m_ErrorFlags & DTAPI_LNB_VOLT_LOW)
    printf("LNB low voltage\n");
```

7.4. Handling LNB Power Status Changes

The **DTA-2132** allows you to register callback functions to handle demodulator events, such as LNB power status changes.

The example below shows a class implementation that handles the LNB power status change event and prints detected errors.

```
// Example implementation class handling the LNB power status changes.
class LnbStatusChangeHandler : public IDtDemodEvent
{
public:
    virtual void LnbPowerStatusHasChanged(const DtLnbPowerStatus& Status) override;
    LnbStatusChangeHandler(int Port) : PortNr(Port){}
    virtual ~LnbStatusChangeHandler() = default;
private:
    int PortNr;
};

// Example implementation of LnbPowerStatusHasChanged callback function
void LnbStatusChangeHandler::LnbPowerStatusHasChanged(const DtLnbPowerStatus& Status)
{
    int ErrorFlags = Status.m_ErrorFlags;
    if (!ErrorFlags)
        printf("Port[%d]: LNB OK\n", PortNr);
    if (ErrorFlags & DTAPI_LNB_POWER_OVL)
        printf("Port[%d]: LNB overload\n", PortNr);
    if (ErrorFlags & DTAPI_LNB_THERM_OHT)
        printf("Port[%d]: LNB overheated\n", PortNr);
    if (ErrorFlags & DTAPI_LNB_VOLT_LOW)
        printf("Port[%d]: LNB low voltage\n", PortNr);
}

// Code to attach the DtInpChannel object goes here.
:  :  :

// Create a handler for LNB-power status changes and register the callback function.
LnbStatusChangeHandler LnbStatusChange(Inp.m_HwFuncDesc.m_Port);
Inp.RegisterDemodCallback(&LnbStatusChange, DTAPI_EV_LNB_POWER_STATUS_CHANGED);

:  :  :

// Use NULL to stop handling events.
Inp.RegisterDemodCallback(NULL);
```