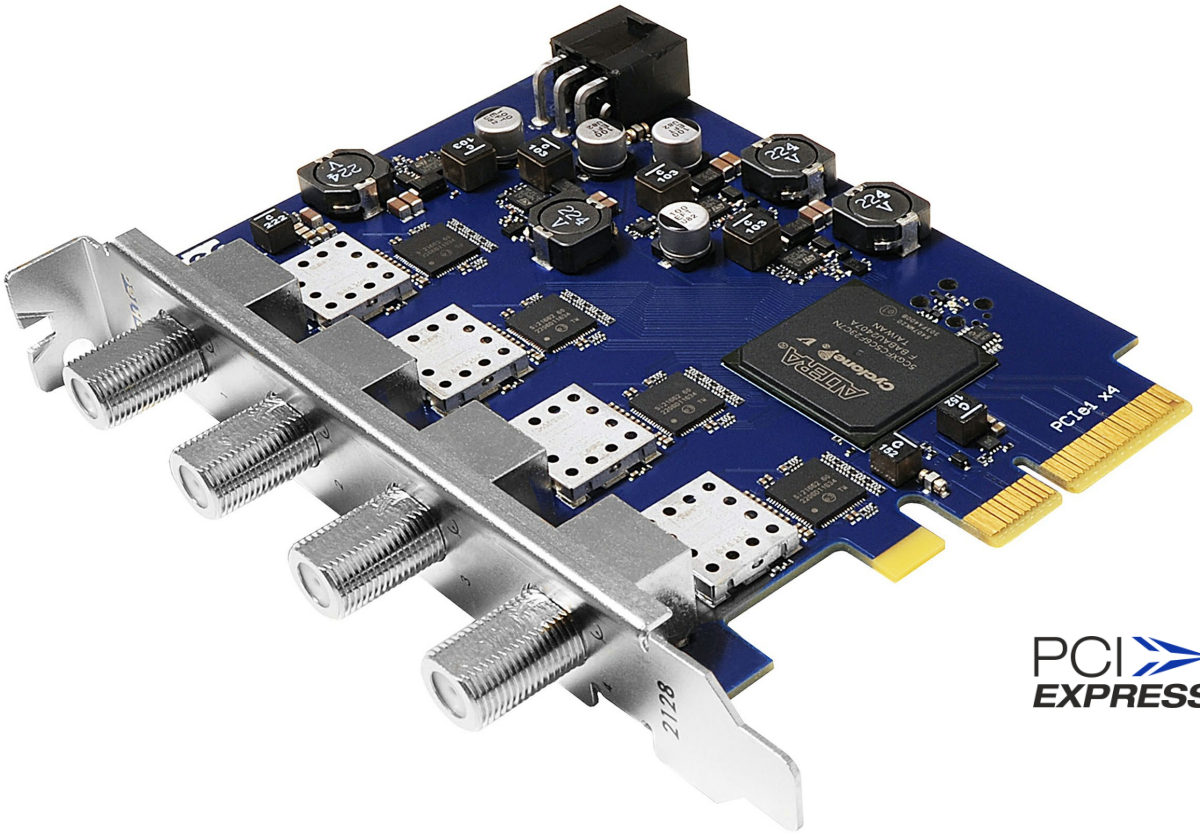
DTA-2128

Octal DVB-S2X Receiver for PCIe



PCI EXPRESS

DATASHEET

Januray 2025

Table of Contents

1. Introduction

1.1. General Description

The **DTA-2128** is DekTec's highest-density satellite receiver on a low-profile PCI Express card (PCIe1 x4). It features four independent L band inputs, each capable of receiving and demodulating two independent satellite signals. This enables the reception of up to eight satellite channels simultaneously. Each receiver is agile in the range of 950 to 2150MHz.

The card supports reception of DVB-S, DVB-S2 and DVB-S2X modulated signals carrying either an MPEG-2 Transport Stream or GSE (Generic Stream Encapsulation) packets. Note that the DTA-2128 does not interpret the content of the MPEG-2 TS or GSE packets, allowing it to receive MPEG-2, H.264, HEVC, or non-video data such as IP traffic, provided the content is encapsulated in a valid Transport Stream or GSE format.

The DTA-2128 can power and control up to four devices, such as LNBs or other compatible equipment, by applying the required voltage to each input connector, which is then delivered through the connected RF cable along with DiSEqC control signals.

Note:

- Simultaneous operation of all four LNBs at full power capacity requires more than the 25W supplied by a standard PCIe slot. This additional power can come from either a high-power PCIe slot capable of delivering 75W or by the external power connector on the DTA-2128. The external-power connector, commonly used for powering GPU cards, supplements the PCIe power.

2. Functional Description

2.1. Block Diagram

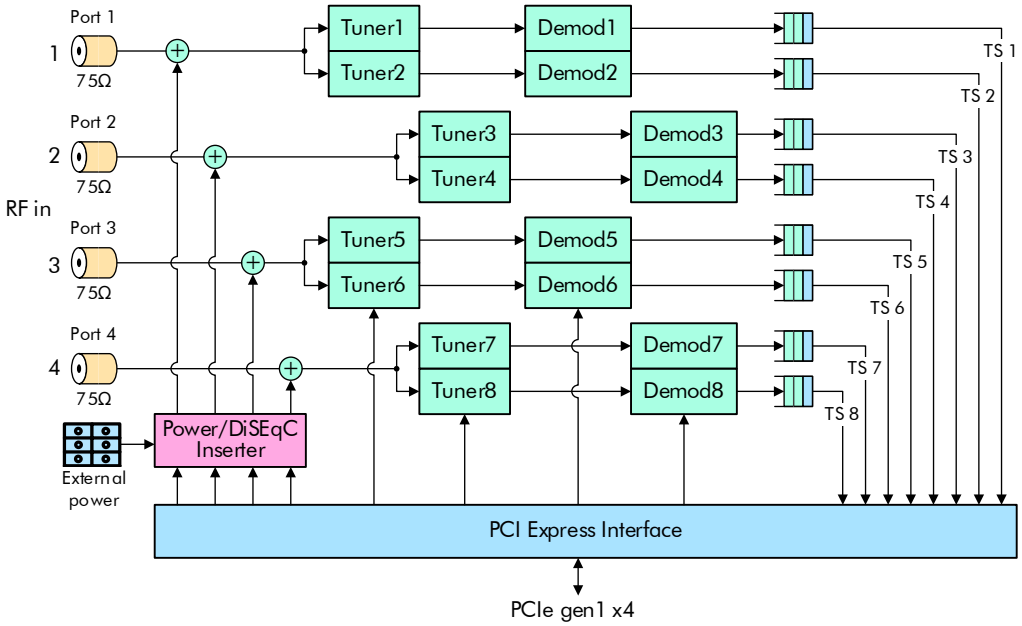


Figure 1. Functional block diagram of the DTA-2128

2.2. Receiving Satellite Signals

The L-band signal from the LNB must be connected to RF input ports (1, 2, 3 or 4). The user can configure the center frequency of the desired signal via software. The **DTA-2128** includes eight tuners, organized as follows:

- **Tuners 1 and 2** are connected to RF input port 1.
- **Tuners 3 and 4** are connected to RF input port 2.
- **Tuners 5 and 6** are connected to RF input port 3.
- **Tuners 7 and 8** are connected to RF input port 4.

Each tuner operates independently and can tune to either the same frequency or different frequencies compared to other tuners. The tuner also provides signal level measurements to the software.

When a signal is locked, the tuner provides the I/Q signal to the demodulator. The demodulator either automatically locks to the signal or uses parameters specified by the software to achieve lock. The demodulator reports the lock status and RF quality metrics back to the software.

2.3. Controlling the LNB

The **DTA-2128** 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.

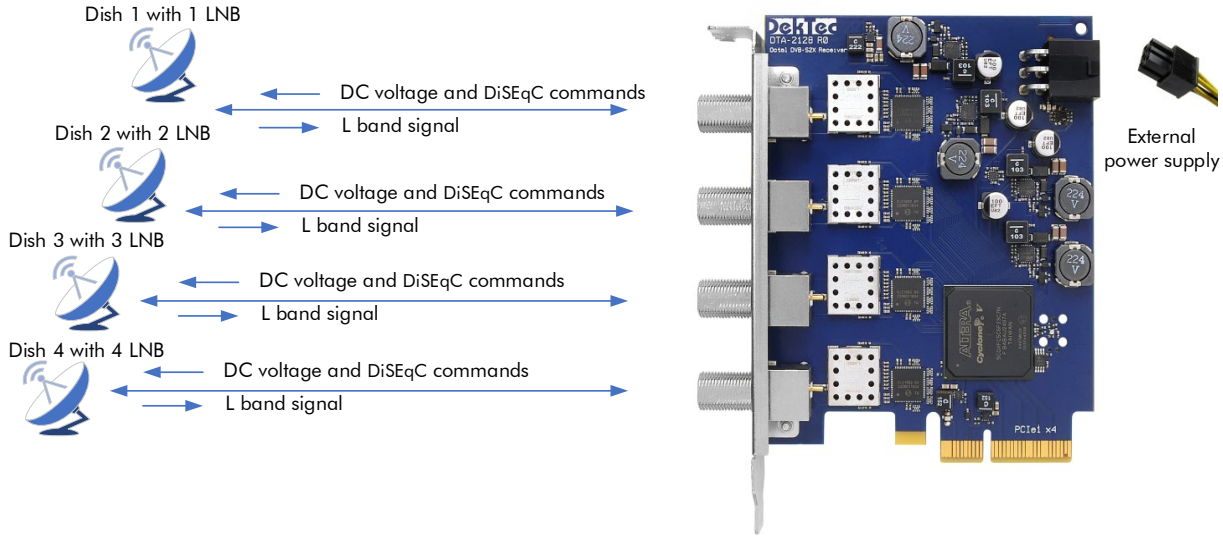


Figure 2. The DTA-2128 can receive a signal while powering/controlling the LNB over the same cable.

Notes

- When powering multiple LNBs, the **DTA-2128** may require external power in addition to the PCIe power, to supplement the power available from the PCIe slot.
- Each RF input port operates independently in terms of powering the LNB and sending DiSEqC commands.
- If the **DTA-2128** is connected to an RF distribution system, LNB power from the card is not required, as the distribution system manages both the power and control of the LNBs.

2.4. Software Support

The DTA-2128 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 one or multiple satellite signals.

The device driver implements low-level operations that require direct access to the DTA-2128 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-2128:

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

3. Specifications

3.1. RF Inputs

The characteristics of the two RF inputs are specified in the table below.

Parameter	Qualification	Min	Typ	Max	Unit
RF INPUT PORT 1, 2, 3, 4					
Connector type			BNC, female		
Impedance			75		Ω
Return loss	950 .. 2150MHz		-10		dB
TUNING					
Frequency range		950		2150	MHz
Sensitivity		-70		-20	dBm
STANDARDS					
DVB-S			EN 300		
DVB-S2			EN 302 307		
DVB-S2X			EN 302 307-2		
Symbol rate*	QPSK, 8-PSK	1		54	MBd
	16-APSK	1		50.7	MBd
	32-APSK	1		40.5	MBd
RF LEVEL MEASUREMENT					
Range		-70		-20	dBm
Accuracy			±3		dBm
MER MEASUREMENT					
Range		0		22	dB
Accuracy			±2		dB

* Section 3.2 lists all supported modulation parameters and the corresponding symbol-rate range.

3.2. Supported Modulation Parameters

The table below specifies the modulation standards, modes, code rates and symbol rates that the DTA-2128 can properly receive.

			Symbol Rate	
Standard	Modulation	Code rate	Min (MBd)	Max (MBd)
DVB-S	QPSK	1/2, 2/3, 3/4, 5/6, 7/8	1.0	54.0
DVB-S2	QPSK	1/2, 2/3, 3/4, 3/5, 4/5, 5/6, 8/9, 9/10	1.0	54.0
	8-PSK	2/3, 3/4, 3/5, 5/6, 8/9, 9/10	1.0	54.0
	16-APSK	2/3, 3/4, 4/5, 5/6, 8/9, 9/10	1.0	50.7
	32-APSK	3/4, 4/5, 5/6, 8/9	1.0	40.6
		9/10	1.0	40.5
DVB-S2X	QPSK	13/45, 9/20, 11/20	1.0	54.0
	8-APSK-L	5/9, 26/45	1.0	54.0
	8-PSK	23/36, 25/36, 13/18	1.0	54.0
	16-APSK-L	1/2, 8/15, 5/9, 3/5, 2/3	1.0	50.7
	16-APSK	26/45, 3/5, 28/45, 23/36, 13/18, 7/9, 77/90	1.0	50.7
	32-APSK-L	2/3	1.0	40.6
	32-APSK	32/45, 11/15, 7/9	1.0	40.6

3.3. Controlling the DISH, LNB and Upstream Equipment

The DTA-2128 provides both power and control to the dish, LNB and/or other upstream equipment through its RF input ports. Power for the LNB can be sourced either from the PCIe bus or via an external power connector located on the DTA-2128. This external connector allows additional power to be supplied directly to the card and is typically connected to the PC's power supply.

The LNB power mode (**Low** or **High**) depends on the available PCIe slot power and whether external power is connected, as summarized below:

		LNB		
PCIe Slot Power	External Power	Power Mode	Typical Current	Note
25W	No	**Low**	225mA	Standard-power slot without external power connected. LNBs draw power from the PCIe slot with a limit of 225mA each.
75W	No	**High**	450mA	High-power slot without external power connected. LNBs draw power from the PCIe slot with a limit of 400mA each.
25/75W	Yes			Standard or high-power slot with external power connected. LNBs draw power from the external power supply with a limit of 450mA each.

Power Mode Query

Software can query the available PCIe slot power and the presence of external power using the **DtInputChannel::LnbPowerStatus** SDK function (see §5.3 for more details). Additionally, Windows utility **DtInfo** reports both values.

LNB Short Circuit Protection

If one of the LNBs experiences a short circuit, the DTA-2128 will immediately stop supplying power to the affected LNB. The SDK will report the short circuit condition to the software.

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-2128 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	12.5	13.0	13.5	V
	Vertical 14V	13.5	14.0	14.5	V
	Horizontal 18V	17.5	18.0	18.5	V
	Horizontal 19V	17.5	19.0	19.5	V
Current	High power mode			450	mA
	Low power mode			225	mA
Power - High power mode	Vertical 13V			5.9	W
	Vertical 14V			6.3	W
	Horizontal 18V			8.1	W
	Horizontal 19V			8.6	W
Power - Low power mode	Vertical 13V			3.3	W
	Vertical 14V			3.5	W
	Horizontal 18V			4.5	W
	Horizontal 19V			4.8	W
Overcurrent Protection					
Current Threshold	High power mode	490	495	500	mA
	Low power mode	240	245	250	mA

3.3.2. DiSEqC – Digital Satellite Equipment Control

DiSEqC (https://en.wikipedia.org/wiki/DiSEqC) is a communication protocol that allows the DTA-2128 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-2128.

Parameter	Qualification	Min	Typ	Max	Unit
DiSEqC					
Version			v1.x*, v2.x**		
Connector type			RF, female		
Amplitude	22kHz tone	0.50	0.65	0.80	mV

* 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		8.0		W
	Running without LNB		9.0		W
	Running four LNBs @200mA		27		W
	Running four LNBs @400mA		45		W
PCI EXPRESS BUS					
Label			PCIe1 x4		
Profile			Low profile		
MECHANICAL					
Dimensions	L x H x D card		85 x 112 x 15		mm
	L x H x D with bracket		111 x 126 x 18		mm
Weight			142		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 RF input ports (1, 2, 3, and 4) of the DTA-2128 exhibit similar performance. For simplicity, measurements in this section are presented for a single port only.

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-2128. 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-S	QPSK	5/6	35%	long	on or off	25.7	25.7
B	DVB-S2	QPSK	1/2	35%	long	on or off	26.2	26.2
C		8-APSK	3/4	25%	long	on or off	27.7	27.7
D		16-APSK	5/6	35%	long	on or off	30.0	30.0
E		32-APSK	8/9	25%	long	on or off	27.7	27.7
F	DVB-S2X	8-APSK-L	5/9	5%	long	on or off	28.2	28.2
G		16-APSK-L	3/5	10%	long	on or off	27.5	27.5
H		32-APSK	32/45	15%	long	on or off	27.7	27.7
I		32-APSK-L	2/3	20%	long	on or off	28.2	28.2

Figure 3. SNR and MER measurements at 1250MHz.

4.2. RF Input Port - Return Loss

The figure below shows the return loss measured at RF input port 1 of the DTA-2128.

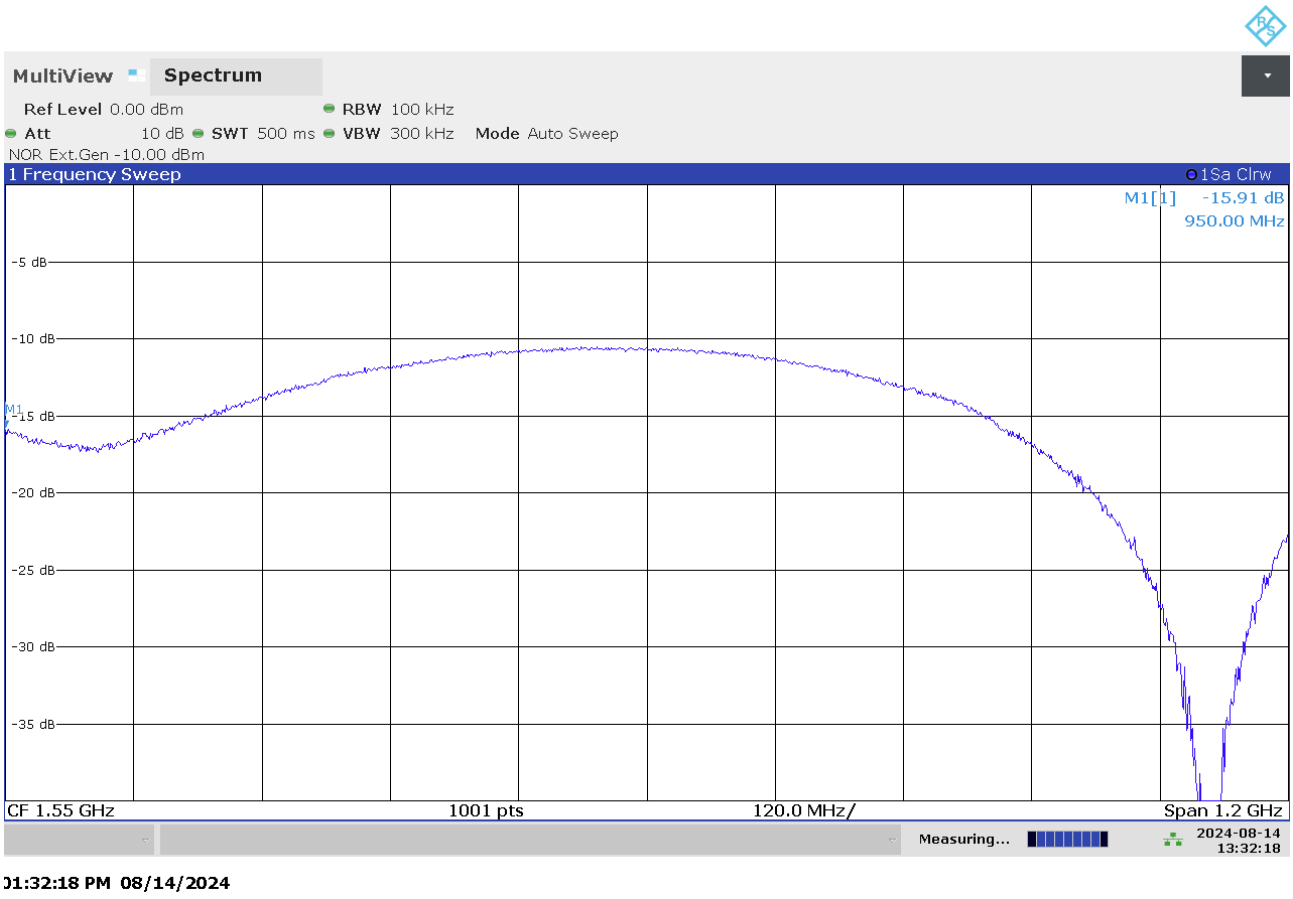


Figure 4. Return loss measurement for RF input 1.

5. Developing Custom Applications with the DTA-2128 and DTAPI

Developers can create custom applications for the DTA-2128 using the free SDK provided by DekTec. The SDK is available for both Windows and Linux and includes a device driver and the DTAPI library.

The device driver implements low-level operations that require direct access to the DTA-2128 hardware, such as initiating and coordinating DMA transfers, handling interrupts, and reading and writing Vital Product Data (VPD). Developers don't need to be aware of the driver functions as they 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". However, additional steps are required when working with the DTA-2128 demodulator card.

5.1. Receiving Data

To receive data from the DTA-2128, 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);
}

// 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 to ensure DTAPI functions operate correctly and handle errors appropriately. of any methods to ensure that DTAPI operates as expected.

5.2. Powering and Controlling LNBs

The DTA-2128 can power an LNB or other compatible equipment through its input connector. To do so, the LNB power must be enabled using `DtInpChannel` method `LnbEnable`. An optional parameter can specify the desired maximum LNB current.

The example below demonstrates enabling LNB power in three different ways, setting the LNB voltage, and sending a DiSEqC switching command.

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

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

// Option 1. Enable the LNB with automatic current management.
// If sufficient board power is available, the maximum LNB current is 400mA,
// otherwise, the LNB current will be limited to 200mA.
Inp.LnbEnable(true);

// Option 2. Enable LNB power with a fixed current limit of 200mA.
Inp.LnbEnable(true, DtLnbPower::LNB_200MA);

// Option 3. Enable LNB power with a fixed current limit of 400mA.
// If insufficient power is available, DTAPI_E_NO_POWER is returned.
Inp.LnbEnable(true, DtLnbPower::LNB_400MA);

// Set LNB voltage to 18 volts.
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));
```

5.3. Retrieving the LNB Power Status

The DTA-2128 provides the ability to retrieve the LNB power status, including the maximum available LNB current and any detected errors, using `DtInpChannel` method `LnbGetPowerStatus`.

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 LNB power mode.
if (Status.m_MaxPower == DtLnbPower::LNB_400MA)
    printf("LNB in high power mode (max 400mA)\n");
else
    printf("LNB in low power mode (max 200mA)\n");

// Print information about the external power connector.
printf("DTA-2128 external power supply: %s; PCIe slot power: %dWatt\n",
                                            Status.m_UsingExtPower ? "yes" : "no",
                                            Dvc.m_DvcDesc.m_PcieMaxSlotPower/1000);

// 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");
```

5.4. Handling LNB Power Status Changes

The DTA-2128 allows you to register callback functions to handle demodulator events, such as changes in the LNB power status.

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 the 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);
```