



# **PDM-IR SDK**

## **PDM-IR**

**Version 3.1.8**

### **Software Development Kit Manual**

April, 2020



# Contents

<b>1</b>	<b>PDM-IR Software Development Kit (PDM-IR_SDK).</b>	<b>5</b>
<b>2</b>	<b>Module Index</b>	<b>7</b>
2.1	Modules . . . . .	7
<b>3</b>	<b>Data Structure Index</b>	<b>9</b>
3.1	Data Structures . . . . .	9
<b>4</b>	<b>File Index</b>	<b>11</b>
4.1	File List . . . . .	11
<b>5</b>	<b>Module Documentation</b>	<b>13</b>
5.1	PDM-IR_SDK custom Types . . . . .	13
5.1.1	Detailed Description . . . . .	14
5.1.2	Enumeration Type Documentation . . . . .	14
5.1.2.1	RESULT . . . . .	14
5.1.2.2	TEMP_VALUES . . . . .	14
5.1.2.3	HOLDOFFTYPE . . . . .	14
5.1.2.4	GATEMODE . . . . .	15
5.1.2.5	GATESHAPE . . . . .	15
5.1.2.6	TREDGE . . . . .	15
5.1.2.7	TRFUNCT . . . . .	15
5.1.2.8	OUTMODE . . . . .	16
5.1.2.9	COUNTERSTATUS . . . . .	17
5.1.2.10	SPAD_TYPE . . . . .	17
5.1.2.11	STATUSBITS . . . . .	17
5.1.2.12	STATUSWARMUP . . . . .	17
5.2	Constructor, destructor, error handling . . . . .	18
5.2.1	Detailed Description . . . . .	18
5.2.2	Function Documentation . . . . .	18
5.2.2.1	PDMIR_SearchDevices(UINT16 *arrayDataSN, UINT16 *numDevsFound) . . .	18
5.2.2.2	PDMIR_OpenCommunication(UINT16 serialNumber) . . . . .	18
5.2.2.3	PDMIR_CloseCommunication(UINT16 serialNumber) . . . . .	18

5.3	Set methods	20
5.3.1	Detailed Description	20
5.3.2	Function Documentation	20
5.3.2.1	PDMIR_SetAmplitude(UINT16 serialNumber, UINT16 AmplimV)	20
5.3.2.2	PDMIR_SetTemperature(UINT16 serialNumber, TEMP_VALUES temp)	21
5.3.2.3	PDMIR_SetHoldOff(UINT16 serialNumber, UINT32 holdOff, HOLDOFFTYPE type)	21
5.3.2.4	PDMIR_SetGateMode(UINT16 serialNumber, GATEMODE gm)	22
5.3.2.5	PDMIR_SetGateShape(UINT16 serialNumber, GATESHAPE gs)	22
5.3.2.6	PDMIR_SetTriggerEdge(UINT16 serialNumber, TREDGE TriggerEdge)	22
5.3.2.7	PDMIR_SetTriggerInTh(UINT16 serialNumber, INT16 TriggerTh)	23
5.3.2.8	PDMIR_SetAuxInTh(UINT16 serialNumber, INT16 TriggerTh)	23
5.3.2.9	PDMIR_SetTriggerFunction(UINT16 serialNumber, TRFUNCT TriggerFunction)	23
5.3.2.10	PDMIR_SetFrequency(UINT16 serialNumber, UINT32 frequency)	24
5.3.2.11	PDMIR_SetTon(UINT16 serialNumber, UINT32 tOn)	24
5.3.2.12	PDMIR_SetTTLOut(UINT16 serialNumber, OUTMODE TTLOut)	24
5.3.2.13	PDMIR_SetNIMOut(UINT16 serialNumber, OUTMODE NIMOut)	25
5.3.2.14	PDMIR_SetDelayTriggerIn(UINT16 serialNumber, UINT8 delay)	25
5.3.2.15	PDMIR_SetDelayTriggerInternal(UINT16 serialNumber, UINT8 delay)	26
5.3.2.16	PDMIR_SetDelayAuxIn(UINT16 serialNumber, UINT8 delay)	26
5.3.2.17	PDMIR_SetDelayGate(UINT16 serialNumber, UINT8 delay)	26
5.3.2.18	PDMIR_SetDelayNIMOut(UINT16 serialNumber, UINT8 delay)	27
5.3.2.19	PDMIR_SetDelayTTLOut(UINT16 serialNumber, UINT8 delay)	27
5.3.2.20	PDMIR_SetCounterIntegrationTime(UINT16 serialNumber, UINT16 integrationTime)	27
5.3.2.21	PDMIR_SetCounterAvalancheEnable(UINT16 serialNumber, COUNTERSTATUS Cstatus)	28
5.3.2.22	PDMIR_SetCounterValidGateEnable(UINT16 serialNumber, COUNTERSTATUS Cstatus)	28
5.3.2.23	PDMIR_SetCounterAuxInEnable(UINT16 serialNumber, COUNTERSTATUS Cstatus)	28
5.3.2.24	PDMIR_SetCounterTriggerInEnable(UINT16 serialNumber, COUNTERSTATUS Cstatus)	29
5.3.2.25	PDMIR_SetCounterInternalTriggerEnable(UINT16 serialNumber, COUNTERSTATUS Cstatus)	29
5.3.2.26	PDMIR_SetModuleOnOff(UINT16 serialNumber, STATUSBITS Mstatus)	30
5.4	Get methods	31
5.4.1	Detailed Description	32
5.4.2	Function Documentation	32
5.4.2.1	PDMIR_GetAmplitude(UINT16 serialNumber, UINT16 *ampi)	32
5.4.2.2	PDMIR_GetTemperature(UINT16 serialNumber, TEMP_VALUES *temp)	32
5.4.2.3	PDMIR_GetHoldOff(UINT16 serialNumber, UINT32 *holdOff, HOLDOFFTYPE *type)	33

5.4.2.4	PDMIR_GetGateMode(UINT16 serialNumber, GATEMODE *gate) . . . . .	33
5.4.2.5	PDMIR_GetGateShape(UINT16 serialNumber, GATESHAPE *shape) . . . . .	33
5.4.2.6	PDMIR_GetTriggerEdge(UINT16 serialNumber, TREDGE *edge) . . . . .	34
5.4.2.7	PDMIR_GetTriggerInTh(UINT16 serialNumber, INT16 *TriggerInTh) . . . . .	34
5.4.2.8	PDMIR_GetAuxInTh(UINT16 serialNumber, INT16 *AuxInTh) . . . . .	34
5.4.2.9	PDMIR_GetTriggerFunction(UINT16 serialNumber, TRFUNCT *function) . . . . .	35
5.4.2.10	PDMIR_GetFrequency(UINT16 serialNumber, UINT32 *freq) . . . . .	35
5.4.2.11	PDMIR_GetTon(UINT16 serialNumber, UINT32 *Ton) . . . . .	35
5.4.2.12	PDMIR_GetTTLOut(UINT16 serialNumber, OUTMODE *TTLOut) . . . . .	36
5.4.2.13	PDMIR_GetNIMOut(UINT16 serialNumber, OUTMODE *NIMOut) . . . . .	36
5.4.2.14	PDMIR_GetDelayTriggerIn(UINT16 serialNumber, UINT8 *delay) . . . . .	36
5.4.2.15	PDMIR_GetDelayTriggerInternal(UINT16 serialNumber, UINT8 *delay) . . . . .	37
5.4.2.16	PDMIR_GetDelayAuxIn(UINT16 serialNumber, UINT8 *delay) . . . . .	37
5.4.2.17	PDMIR_GetDelayGate(UINT16 serialNumber, UINT8 *delay) . . . . .	38
5.4.2.18	PDMIR_GetDelayNIMOut(UINT16 serialNumber, UINT8 *delay) . . . . .	38
5.4.2.19	PDMIR_GetDelayTTLOut(UINT16 serialNumber, UINT8 *delay) . . . . .	38
5.4.2.20	PDMIR_GetCounterIntegrationTime(UINT16 serialNumber, UINT16 *integrationTime) . . . . .	39
5.4.2.21	PDMIR_GetCounterAvalancheStatus(UINT16 serialNumber, COUNTERSTATUS *status) . . . . .	39
5.4.2.22	PDMIR_GetCounterValidGateStatus(UINT16 serialNumber, COUNTERSTATUS *status) . . . . .	39
5.4.2.23	PDMIR_GetCounterAuxInStatus(UINT16 serialNumber, COUNTERSTATUS *status) . . . . .	40
5.4.2.24	PDMIR_GetCounterTriggerInStatus(UINT16 serialNumber, COUNTERSTATUS *status) . . . . .	40
5.4.2.25	PDMIR_GetCounterInternalTriggerStatus(UINT16 serialNumber, COUNTERSTATUS *status) . . . . .	40
5.4.2.26	PDMIR_GetAllCountersStatus(UINT16 serialNumber, COUNTERSTATUS *avalancheStatus, COUNTERSTATUS *validGateStatus, COUNTERSTATUS *auxInStatus, COUNTERSTATUS *triggerInStatus, COUNTERSTATUS *internalTriggerStatus) . . . . .	41
5.4.2.27	PDMIR_GetCounterAvalancheLastValue(UINT16 serialNumber, UINT8 *index, UINT32 *counts) . . . . .	41
5.4.2.28	PDMIR_GetCounterValidGateLastValue(UINT16 serialNumber, UINT8 *index, UINT32 *counts) . . . . .	42
5.4.2.29	PDMIR_GetCounterAuxInLastValue(UINT16 serialNumber, UINT8 *index, UINT32 *counts) . . . . .	42
5.4.2.30	PDMIR_GetCounterTriggerInLastValue(UINT16 serialNumber, UINT8 *index, UINT32 *counts) . . . . .	43
5.4.2.31	PDMIR_GetCounterInternalTriggerLastValue(UINT16 serialNumber, UINT8 *index, UINT32 *counts) . . . . .	43
5.4.2.32	PDMIR_GetAllCountersLastValue(UINT16 serialNumber, UINT8 *index, UINT32 *avalancheCounts, UINT32 *validGateCounts, UINT32 *auxInCounts, UINT32 *triggerInCounts, UINT32 *internalTriggerCounts) . . . . .	43

5.4.2.33	PDMIR_GetCounterAvalancheValueAtIndex(UINT16 serialNumber, UINT8 *index, UINT32 *counts)	44
5.4.2.34	PDMIR_GetCounterValidGateValueAtIndex(UINT16 serialNumber, UINT8 *index, UINT32 *counts)	44
5.4.2.35	PDMIR_GetCounterAuxInValueAtIndex(UINT16 serialNumber, UINT8 *index, UINT32 *counts)	45
5.4.2.36	PDMIR_GetCounterTriggerInValueAtIndex(UINT16 serialNumber, UINT8 *index, UINT32 *counts)	45
5.4.2.37	PDMIR_GetCounterInternalTriggerValueAtIndex(UINT16 serialNumber, UINT8 *index, UINT32 *counts)	46
5.4.2.38	PDMIR_GetAllCountersValueAtIndex(UINT16 serialNumber, UINT8 *index, UINT32 *avalancheCounts, UINT32 *validGateCounts, UINT32 *auxInCounts, UINT32 *triggerInCounts, UINT32 *internalTriggerCounts)	46
5.4.2.39	PDMIR_GetModuleInfo(UINT16 serialNumber, MODULEINFO *info)	47
5.4.2.40	PDMIR_GetModuleStatus(UINT16 serialNumber, STATUSBITS *mStatus, STATUSBITS *mTemperature, STATUSBITS *mSpad, STATUSBITS *mGate, STATUSWARMUP *mWarm, UINT32 *errors)	47
5.4.2.41	PDMIR_ErrorTranslator(UINT32 error, char *stringOut)	47
5.5	Configuration methods	49
5.5.1	Detailed Description	49
5.5.2	Function Documentation	49
5.5.2.1	PDMIR_SaveCurrentConfig(UINT16 serialNumber, UINT8 configNumber, char *configName)	49
5.5.2.2	PDMIR_SaveCurrentConfigForceOverWrite(UINT16 serialNumber, UINT8 configNumber, char *configName)	49
5.5.2.3	PDMIR_DeleteConfig(UINT16 serialNumber, UINT8 configNumber)	50
5.5.2.4	PDMIR_SetConfigPowerUp(UINT16 serialNumber, UINT8 configNumber)	50
5.5.2.5	PDMIR_LoadConfig(UINT16 serialNumber, UINT8 configNumber)	51
5.5.2.6	PDMIR_GetCurrentConfig(UINT16 serialNumber, MODULECONFIG *config)	51
5.5.2.7	PDMIR_GetConfigX(UINT16 serialNumber, UINT8 configNumber, MODULECONFIG *config, char *configName)	51
5.5.2.8	PDMIR_GetConfigPowerUp(UINT16 serialNumber, UINT8 *configNumber)	52
<b>6</b>	<b>Data Structure Documentation</b>	<b>53</b>
6.1	MODULECONFIG Struct Reference	53
6.1.1	Detailed Description	53
6.2	MODULEINFO Struct Reference	54
6.2.1	Detailed Description	54
<b>7</b>	<b>File Documentation</b>	<b>55</b>
7.1	PDM-IR_SDK.h File Reference	55
7.1.1	Detailed Description	58

## Chapter 1

# PDM-IR Software Development Kit (PDM-IR\_SDK).

The MPD PDM-IR is based on a InGaAs/InP SPAD for the detection of near-infrared single photons up to 1700 nm. The module includes a pulse generator for gating the detector, a front-end circuit for avalanche sensing and a fast circuitry for detector quenching and resetting. The internal counters monitor the major signals (photons, internal trigger, trigger in, Aux In, valid gate).

**IMPORTANT** In order to execute a program which links to the SDK libraries, the following file is required:

PDM-IR\_SDK.dll            Software development kit library





## Chapter 2

# Module Index

### 2.1 Modules

Here is a list of all modules:

PDM-IR_SDK custom Types . . . . .	13
Constructor, destructor, error handling . . . . .	18
Set methods . . . . .	20
Get methods . . . . .	31
Configuration methods . . . . .	49



## Chapter 3

# Data Structure Index

### 3.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">_EVENT_HANDLE</a>	??
<a href="#">_OVERLAPPED</a>	??
<a href="#">_SECURITY_ATTRIBUTES</a>	??
<a href="#">MODULECONFIG</a>	
Module Configuration Structure	53
<a href="#">MODULEINFO</a>	
Structure containing the information of the PDM-IR	54



## Chapter 4

# File Index

### 4.1 File List

Here is a list of all documented files with brief descriptions:

<a href="#">PDM-IR_SDK.h</a>	
PDM-IR software development kit . . . . .	<a href="#">55</a>
<b>WinTypes.h</b> . . . . .	<b>??</b>



## Chapter 5

# Module Documentation

### 5.1 PDM-IR\_SDK custom Types

#### Data Structures

- struct [MODULECONFIG](#)
- struct [MODULEINFO](#)

#### Typedefs

- typedef unsigned char [UINT8](#)
- typedef signed char [INT8](#)
- typedef signed short [INT16](#)
- typedef unsigned short [UINT16](#)
- typedef unsigned int [UINT32](#)
- typedef signed int [INT32](#)

#### Enumerations

- enum [RESULT](#) {  
    [RESULT\\_OK](#), [RESULT\\_DATA\\_OUT\\_LIMIT\\_HIGH](#), [RESULT\\_DATA\\_OUT\\_LIMIT\\_LOW](#), [RESULT\\_DATA\\_ERROR](#),  
    [RESULT\\_ERROR\\_COMMUNICATION](#) }
- enum [TEMP\\_VALUES](#) { [THIGH](#) = 0, [TMEDIUM](#) = 1, [TLOW](#) = 2, [TLOWEST](#) = 3 }
- enum [HOLDOFFTYPE](#) { [HOLDOFFTYPE\\_EDGE](#) = 0, [HOLDOFFTYPE\\_LEVEL](#) = 1 }
- enum [GATEMODE](#) { [GATEMODE\\_INTERNAL](#) = 0, [GATEMODE\\_EXTERNAL](#) = 1 }
- enum [GATESHAPE](#) { [GATESHAPE\\_FIXEDGATE](#) = 0, [GATESHAPE\\_FREEGATE](#) = 1, [GATESHAPE\\_FREERUNNING](#) = 2 }
- enum [TREDGE](#) { [TREDGE\\_LH](#) = 0, [TREDGE\\_HL](#) = 1 }
- enum [TRFUNCT](#) {  
    [In\\_AND\\_Aux](#) = 0, [In\\_OR\\_Aux](#) = 1, [In\\_XOR\\_Aux](#) = 2, [notIn\\_AND\\_Aux](#) = 3,  
    [notIn\\_OR\\_Aux](#) = 4, [notIn\\_XOR\\_Aux](#) = 5, [In\\_AND\\_notAux](#) = 6, [In\\_OR\\_notAux](#) = 7,  
    [In\\_XOR\\_notAux](#) = 8, [notIn\\_AND\\_notAux](#) = 9, [notIn\\_OR\\_notAux](#) = 10, [notIn\\_XOR\\_notAux](#) = 11,  
    [In\\_NAND\\_Aux](#) = 12, [In\\_NOR\\_Aux](#) = 13, [In\\_XNOR\\_Aux](#) = 14, [notIn\\_NAND\\_Aux](#) = 15,  
    [notIn\\_NOR\\_Aux](#) = 16, [notIn\\_XNOR\\_Aux](#) = 17, [In\\_NAND\\_notAux](#) = 18, [In\\_NOR\\_notAux](#) = 19,  
    [In\\_XNOR\\_notAux](#) = 20, [notIn\\_NAND\\_notAux](#) = 21, [notIn\\_NOR\\_notAux](#) = 22, [notIn\\_XNOR\\_notAux](#) = 23,  
    [only\\_notIn](#) = 24, [only\\_IN](#) = 25 }
- enum [OUTMODE](#) {  
    [OUTMODE\\_TRIGGERGATE](#) = 0, [OUTMODE\\_PHOTONOUT](#) = 1, [OUTMODE\\_HOLDOFF](#) = 2, [OUTMODE\\_VALIDGATE](#) = 3,  
    [OUTMODE\\_INTERNALTRIGGER](#) = 4, [OUTMODE\\_MODULESTATUS](#) = 5 }

- enum **COUNTERSTATUS** { **COUNTERSTATUS\_OFF** = 0, **COUNTERSTATUS\_ON** = 1, **COUNTERSTATUS\_Continuous** = 2 }
- enum **SPAD\_TYPE** { **TYPE\_1** = 0x1, **TYPE\_2** = 0x2, **TYPE\_3** = 0x3, **TYPE\_4** = 0x4, **TYPE\_5** = 0x5, **TYPE\_6** = 0x6, **TYPE\_7** = 0x7 }
- enum **STATUSBITS** { **STATUSBITS\_OFF** = 0, **STATUSBITS\_ON** = 1, **STATUSBITS\_Updating** = 2, **STATUSBITS\_SafeValue** = 3, **STATUSBITS\_ToBeUpdated** = 4, **STATUSBITS\_Updated** = 5 }
- enum **STATUSWARMUP** { **STATUSWARMUP\_isWarmingUp** = 0, **STATUSWARMUP\_isSteadyState** = 1 }

### 5.1.1 Detailed Description

Custom types used by the SDK.

### 5.1.2 Enumeration Type Documentation

#### 5.1.2.1 enum RESULT

Error table enum.

Error code returned by the PDM-IR functions.

Enumerator

**RESULT\_OK** result as expected. No errors.

**RESULT\_DATA\_OUT\_LIMIT\_HIGH** At least one of the input values are above the maximum value accepted.

**RESULT\_DATA\_OUT\_LIMIT\_LOW** At least one of the input values are below the minimum value accepted.

**RESULT\_DATA\_ERROR** Error parsing the values referenced.

**RESULT\_ERROR\_COMMUNICATION** Error communicating with the module.

#### 5.1.2.2 enum TEMP\_VALUES

Temperature enum.

Temperature of the SPAD.

Enumerator

**THIGH** High temperature: the dark count rate is the highest than other temperature modes, but no particular caution is required for module thermal stability.

**TMEDIUM** Medium temperature: the dark count rate is the one shown in the test report. In this case the module requires to be placed on a reasonable heat sink for uninterrupted operation.

**TLOW** Low temperature: the dark count rate is very low. In this case the module requires a very good heat sink for optimal thermal dissipation and uninterrupted operation.

**TLOWEST** Lowest temperature: the dark count rate is the lowest as possible. In this case the module requires a very good heat sink for thermal dissipation and a fan, in order to work without errors. Also, a low ambient temperature (of about 20-22C) will be required.

#### 5.1.2.3 enum HOLDOFFTYPE

hold Off Type enum.

The enum select the PDM-IR hold off type.



## Enumerator

**HOLDOFFTYPE\_EDGE** When the hold off time is over, the gate is not applied until the next rising edge of the gate window.

**HOLDOFFTYPE\_LEVEL** The Gate window is applied after the hold off time is over. This type is required in free running mode

## 5.1.2.4 enum GATEMODE

Gate Mode enum.

The enum select the PDM-IR internal or external trigger. The chose trigger is in logic function with the Aux In

See also

[TRFUNCT](#)

## Enumerator

**GATEMODE\_INTERNAL** The Gate trigger is the internal trigger.

**GATEMODE\_EXTERNAL** The Gate trigger is the external trigger.

## 5.1.2.5 enum GATESHAPE

Gate Shape enum.

The enum specifies if the PDM-IR has a fixed width gate or the same width of the trigger gate, or is in free running mode.

## Enumerator

**GATESHAPE\_FIXEDGATE** The Gate has a fixed width determinated by the Ton.

**GATESHAPE\_FREEGATE** The Gate has the same digital shape and duration of the trigger gate signal.

**GATESHAPE\_FREERUNNING** The SPAD is always on when not in hold off.

## 5.1.2.6 enum TREDGE

Edge Type enum.

it selects the Edge sensitivity of the input.

## Enumerator

**TREDGE\_LH** Rising Edge Trigger.

**TREDGE\_HL** Falling Edge Trigger.

## 5.1.2.7 enum TRFUNCT

Function Type enum.

Function between the trigger (Internal Trigger or Trigger In chosen by GATEMODE) and the Aux In.

See also

[GATEMODE](#)

Enumerator

***In\_AND\_Aux*** Selected trigger AND Aux In.  
***In\_OR\_Aux*** Selected trigger OR Aux In.  
***In\_XOR\_Aux*** Selected trigger XOR Aux In.  
***notIn\_AND\_Aux*** Inverted Selected trigger AND Aux In.  
***notIn\_OR\_Aux*** Inverted Selected trigger OR Aux In.  
***notIn\_XOR\_Aux*** Inverted Selected trigger XOR Aux In.  
***In\_AND\_notAux*** Selected trigger AND Inverted Aux In.  
***In\_OR\_notAux*** Selected trigger OR Inverted Aux In.  
***In\_XOR\_notAux*** Selected trigger XOR Inverted Aux In.  
***notIn\_AND\_notAux*** Inverted Selected trigger AND Inverted Aux In.  
***notIn\_OR\_notAux*** Inverted Selected trigger OR Inverted Aux In.  
***notIn\_XOR\_notAux*** Inverted Selected trigger XOR Inverted Aux In.  
***In\_NAND\_Aux*** Selected trigger NAND Aux In.  
***In\_NOR\_Aux*** Selected trigger NOR Aux In.  
***In\_XNOR\_Aux*** Selected trigger XNOR Aux In.  
***notIn\_NAND\_Aux*** Inverted Selected trigger NAND Aux In.  
***notIn\_NOR\_Aux*** Inverted Selected trigger NOR Aux In.  
***notIn\_XNOR\_Aux*** Inverted Selected trigger XNOR Aux In.  
***In\_NAND\_notAux*** Selected trigger NAND Inverted Aux In.  
***In\_NOR\_notAux*** Selected trigger NOR Inverted Aux In.  
***In\_XNOR\_notAux*** Selected trigger XNOR Inverted Aux In.  
***notIn\_NAND\_notAux*** Inverted Selected trigger NAND Inverted Aux In.  
***notIn\_NOR\_notAux*** Inverted Selected trigger NOR Inverted Aux In.  
***notIn\_XNOR\_notAux*** Inverted Selected trigger XNOR Inverted Aux In.  
***only\_notIn*** Inverted Selected trigger.  
***only\_IN*** Selected trigger.

#### 5.1.2.8 enum OUTMODE

Output Type enum.

it specifies the signal out from TLL out or NIM out.

Enumerator

***OUTMODE\_TRIGGERGATE*** The Output reproduces the trigger gate signal.  
***OUTMODE\_PHOTONOUT*** The Output reproduces the photon out signal.  
***OUTMODE\_HOLDOFF*** The Output reproduces the hold off signal.  
***OUTMODE\_VALIDGATE*** The Output reproduces the valid gate signal.  
***OUTMODE\_INTERNALTRIGGER*** The Output reproduces the internal trigger signal.  
***OUTMODE\_MODULESTATUS*** The Output is high in case of error of the module (electrical signal of the RED led)

## 5.1.2.9 enum COUNTERSTATUS

Counter Status enum.

It indicates the counter status.

Enumerator

- COUNTERSTATUS\_OFF** The Counter is disabled.
- COUNTERSTATUS\_ON** The Counter is enabled only for one period.
- COUNTERSTATUS\_Continuous** The Counter is enabled.

## 5.1.2.10 enum SPAD\_TYPE

SPAD Type enum.

The SPAD type used in the module.

Enumerator

- TYPE\_1** useful parameter for device statistics. Type1.
- TYPE\_2** useful parameter for device statistics. Type2.
- TYPE\_3** useful parameter for device statistics. Type3.
- TYPE\_4** useful parameter for device statistics. Type4.
- TYPE\_5** useful parameter for device statistics. Type5.
- TYPE\_6** useful parameter for device statistics. Type6.
- TYPE\_7** useful parameter for device statistics. Type7.

## 5.1.2.11 enum STATUSBITS

Parameter Status enum.

The enum indicates the actual status of a parameter.

Enumerator

- STATUSBITS\_OFF** The Parameter is disabled.
- STATUSBITS\_ON** The Parameter is enabled.
- STATUSBITS\_Updating** The Parameter is updating the state.
- STATUSBITS\_SafeValue** The Parameter has a safe value, used before update the value waiting the right update order.
- STATUSBITS\_ToBeUpdated** The Parameter needs to be updated.
- STATUSBITS\_Updated** The Parameter is updated.

## 5.1.2.12 enum STATUSWARMUP

Warming Up module enum.

The enum indicates if the warm up of the module is completed or not.

Enumerator

- STATUSWARMUP\_isWarmingUp** The module is warming up.
- STATUSWARMUP\_isSteadyState** The module has a stable temperature.

## 5.2 Constructor, destructor, error handling

### Functions

- DIISDKExport [RESULT PDMIR\\_SearchDevices](#) (UINT16 \*arrayDataSN, UINT16 \*numDevsFound)
- DIISDKExport [RESULT PDMIR\\_OpenCommunication](#) (UINT16 serialNumber)
- DIISDKExport [RESULT PDMIR\\_CloseCommunication](#) (UINT16 serialNumber)

#### 5.2.1 Detailed Description

Functions to Search, Open and Close the communication with the PDM-IR.

#### 5.2.2 Function Documentation

##### 5.2.2.1 DIISDKExport [RESULT PDMIR\\_SearchDevices](#) ( [UINT16 \\* arrayDataSN](#), [UINT16 \\* numDevsFound](#) )

Modules list Constructor.

It allocates a memory block to contain the list of the connected and recognized PDM-IR. It scans all the available USB port on the PC and when a PDM-IR is found, it add the device to fill the list.

#### Parameters

<i>arrayDataSN</i>	Pointer to PDM-IR list handle. This parameter is referenced.
<i>numDevsFound</i>	Pointer to number of devices found. This parameter is referenced.

#### Returns

RESULT\_OK The list was successfully created and populated.  
 RESULT\_ERROR\_COMMUNICATION Error checking the device list.  
 RESULT\_DATA\_OUT\_LIMIT\_LOW no devices found.

##### 5.2.2.2 DIISDKExport [RESULT PDMIR.OpenCommunication](#) ( [UINT16 serialNumber](#) )

Open the communication with a module.

It opens the communication and allocates a memory block to manage the connected device, identified by the serial number.

#### Parameters

<i>serialNumber</i>	PDM-IR serial number
---------------------	----------------------

#### Returns

RESULT\_OK The communication between the module and the pc is opened.  
 RESULT\_ERROR\_COMMUNICATION Error opening the communication channel.

##### 5.2.2.3 DIISDKExport [RESULT PDMIR.CloseCommunication](#) ( [UINT16 serialNumber](#) )

Close the communication with a module.

It closes the communication and deallocates the memory. The module is identified by the serial number.

## Parameters

<i>serialNumber</i>	PDM-IR serial number
---------------------	----------------------

## Returns

RESULT\_OK The communication between the module and the pc is correctly closed.  
RESULT\_ERROR\_COMMUNICATION Error closing the communication channel.

## 5.3 Set methods

### Functions

- DIISDKExport [RESULT PDMIR\\_SetAmplitude](#) (UINT16 serialNumber, [UINT16](#) AmplimV)
- DIISDKExport [RESULT PDMIR\\_SetTemperature](#) (UINT16 serialNumber, [TEMP\\_VALUES](#) temp)
- DIISDKExport [RESULT PDMIR\\_SetHoldOff](#) (UINT16 serialNumber, [UINT32](#) holdOff, [HOLDOFFTYPE](#) type)
- DIISDKExport [RESULT PDMIR\\_SetGateMode](#) (UINT16 serialNumber, [GATEMODE](#) gm)
- DIISDKExport [RESULT PDMIR\\_SetGateShape](#) (UINT16 serialNumber, [GATESHAPE](#) gs)
- DIISDKExport [RESULT PDMIR\\_SetTriggerEdge](#) (UINT16 serialNumber, [TREDGE](#) TriggerEdge)
- DIISDKExport [RESULT PDMIR\\_SetTriggerInTh](#) (UINT16 serialNumber, [INT16](#) TriggerTh)
- DIISDKExport [RESULT PDMIR\\_SetAuxInTh](#) (UINT16 serialNumber, [INT16](#) TriggerTh)
- DIISDKExport [RESULT PDMIR\\_SetTriggerFunction](#) (UINT16 serialNumber, [TRFUNCT](#) TriggerFunction)
- DIISDKExport [RESULT PDMIR\\_SetFrequency](#) (UINT16 serialNumber, [UINT32](#) frequency)
- DIISDKExport [RESULT PDMIR\\_SetTon](#) (UINT16 serialNumber, [UINT32](#) tOn)
- DIISDKExport [RESULT PDMIR\\_SetTTLOut](#) (UINT16 serialNumber, [OUTMODE](#) TTLOut)
- DIISDKExport [RESULT PDMIR\\_SetNIMOut](#) (UINT16 serialNumber, [OUTMODE](#) NIMOut)
- DIISDKExport [RESULT PDMIR\\_SetDelayTriggerIn](#) (UINT16 serialNumber, [UINT8](#) delay)
- DIISDKExport [RESULT PDMIR\\_SetDelayTriggerInternal](#) (UINT16 serialNumber, [UINT8](#) delay)
- DIISDKExport [RESULT PDMIR\\_SetDelayAuxIn](#) (UINT16 serialNumber, [UINT8](#) delay)
- DIISDKExport [RESULT PDMIR\\_SetDelayGate](#) (UINT16 serialNumber, [UINT8](#) delay)
- DIISDKExport [RESULT PDMIR\\_SetDelayNIMOut](#) (UINT16 serialNumber, [UINT8](#) delay)
- DIISDKExport [RESULT PDMIR\\_SetDelayTTLOut](#) (UINT16 serialNumber, [UINT8](#) delay)
- DIISDKExport [RESULT PDMIR\\_SetCounterIntegrationTime](#) (UINT16 serialNumber, [UINT16](#) integrationTime)
- DIISDKExport [RESULT PDMIR\\_SetCounterAvalancheEnable](#) (UINT16 serialNumber, [COUNTERSTATUS](#) Cstatus)
- DIISDKExport [RESULT PDMIR\\_SetCounterValidGateEnable](#) (UINT16 serialNumber, [COUNTERSTATUS](#) Cstatus)
- DIISDKExport [RESULT PDMIR\\_SetCounterAuxInEnable](#) (UINT16 serialNumber, [COUNTERSTATUS](#) Cstatus)
- DIISDKExport [RESULT PDMIR\\_SetCounterTriggerInEnable](#) (UINT16 serialNumber, [COUNTERSTATUS](#) Cstatus)
- DIISDKExport [RESULT PDMIR\\_SetCounterInternalTriggerEnable](#) (UINT16 serialNumber, [COUNTERSTATUS](#) Cstatus)
- DIISDKExport [RESULT PDMIR\\_SetModuleOnOff](#) (UINT16 serialNumber, [STATUSBITS](#) Mstatus)

#### 5.3.1 Detailed Description

Functions to set parameters of the PDM-IR.

#### 5.3.2 Function Documentation

##### 5.3.2.1 DIISDKExport [RESULT PDMIR\\_SetAmplitude](#) ( [UINT16](#) serialNumber, [UINT16](#) AmplimV )

Set the gate amplitude of the PDM-IR.

##### Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>AmplimV</i>	Desired Amplitude in mV. Allowed range: 2000 mV to 7000 mV in fixed gate mode, 2000 mV to 5000 mV in free gate mode and free running, at 100 mV steps.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_OUT\_LIMIT\_HIGH The value set is higher than max value.  
 RESULT\_DATA\_OUT\_LIMIT\_LOW The value set is lower than min value.

## See also

[PDMIR\\_GetGateShape\(\)](#)  
[PDMIR\\_GetAmplitude\(\)](#)

### 5.3.2.2 DIISDKExport RESULT PDMIR.SetTemperature ( UINT16 serialNumber, TEMP\_VALUES temp )

Set the working temperature of the PDM-IR.

## Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>temp</i>	Desired Temperature as TEMP_VALUES.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_OUT\_LIMIT\_HIGH The value set is higher than max value.  
 RESULT\_DATA\_OUT\_LIMIT\_LOW The value set is lower than min value.

## See also

[PDMIR\\_GetTemperature\(\)](#)

### 5.3.2.3 DIISDKExport RESULT PDMIR.SetHoldOff ( UINT16 serialNumber, UINT32 holdOff, HOLDOFFTYPE type )

Set the hold off time of the SPAD.

## Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>holdOff</i>	Desired hold off time in ns. Allowed range: 1000 ns to 3000000 ns, in 100 ns steps.
<i>type</i>	Select edge or level hold off type.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_OUT\_LIMIT\_HIGH The value set is higher than max value.  
 RESULT\_DATA\_OUT\_LIMIT\_LOW The value set is lower than min value.

## See also

[PDMIR\\_GetHoldOff\(\)](#)

### 5.3.2.4 DIISDKExport RESULT PDMIR.SetGateMode ( UINT16 serialNumber, GATEMODE gm )

Set the gate mode of the PDM-IR.

#### Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>gm</i>	Desired gate mode as GATEMODE typedef. It can select internal trigger or trigger in.

#### Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_OUT\_LIMIT\_HIGH The value set is higher than max value.  
 RESULT\_DATA\_OUT\_LIMIT\_LOW The value set is lower than min value.  
 RESULT\_DATA\_ERROR The type is not HOLDOFFTYPE.

#### See also

[PDMIR\\_GetGateMode\(\)](#)

### 5.3.2.5 DIISDKExport RESULT PDMIR.SetGateShape ( UINT16 serialNumber, GATESHAPE gs )

Set the gate shape of the PDM-IR.

#### Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>gs</i>	Desired gate shape as GATESHAPE typedef.

#### Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_OUT\_LIMIT\_HIGH The value set is higher than max value.  
 RESULT\_DATA\_OUT\_LIMIT\_LOW The value set is lower than min value.

#### See also

[PDMIR\\_GetGateShape\(\)](#)

### 5.3.2.6 DIISDKExport RESULT PDMIR.SetTriggerEdge ( UINT16 serialNumber, TREDGE TriggerEdge )

Set the Trigger Edge of the PDM-IR.

#### Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>TriggerEdge</i>	trigger edge as TREDGE typedef.

#### Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_OUT\_LIMIT\_HIGH The value set is higher than max value.



RESULT\_DATA\_OUT\_LIMIT\_LOW The value set is lower than min value.

See also

[PDMIR\\_GetTriggerEdge\(\)](#)

#### 5.3.2.7 DIISDKExport RESULT PDMIR.SetTriggerInTh ( UINT16 serialNumber, INT16 TriggerTh )

Set the Trigger In Threshold of the PDM-IR.

Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>TriggerTh</i>	trigger In Threshold in mV. Allowed range: -2000 mV to 2000 mV, in 10 mV steps.

Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_OUT\_LIMIT\_HIGH The value set is higher than max value.  
 RESULT\_DATA\_OUT\_LIMIT\_LOW The value set is lower than min value.

See also

[PDMIR\\_SetTriggerInTh\(\)](#)

#### 5.3.2.8 DIISDKExport RESULT PDMIR.SetAuxInTh ( UINT16 serialNumber, INT16 TriggerTh )

Set the Aux In Threshold of the PDM-IR.

Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>TriggerTh</i>	Aux In Threshold in mV.

Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR. Allowed range: -2000 mV to 2000 mV, at 10 mV steps.  
 RESULT\_DATA\_OUT\_LIMIT\_HIGH The value set is higher than max value.  
 RESULT\_DATA\_OUT\_LIMIT\_LOW The value set is lower than min value.

See also

[PDMIR\\_GetAuxInTh\(\)](#)

#### 5.3.2.9 DIISDKExport RESULT PDMIR.SetTriggerFunction ( UINT16 serialNumber, TRFUNCT TriggerFunction )

Set the Function between Aux In and trigger (in or internal) of the PDM-IR.

Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>TriggerFunction</i>	trigger function as TRFUNCT typedef.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_OUT\_LIMIT\_HIGH The value set is higher than max value.  
 RESULT\_DATA\_OUT\_LIMIT\_LOW The value set is lower than min value.

## See also

[PDMIR\\_GetTriggerFunction\(\)](#)

#### 5.3.2.10 DIISDKExport RESULT PDMIR\_SetFrequency ( UINT16 *serialNumber*, UINT32 *frequency* )

Set the internal trigger frequency in Hz of the PDM-IR.

## Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>frequency</i>	Internal trigger frequency in Hz. Allowed range: 100 Hz to 100000000 Hz (100 MHz), at 100 Hz steps.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_OUT\_LIMIT\_HIGH The value set is higher than max value.  
 RESULT\_DATA\_OUT\_LIMIT\_LOW The value set is lower than min value.

## See also

[PDMIR\\_GetFrequency\(\)](#)

#### 5.3.2.11 DIISDKExport RESULT PDMIR\_SetTon ( UINT16 *serialNumber*, UINT32 *tOn* )

Set the Ton time in ns of the PDM-IR.

## Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>tOn</i>	gate window time in ns. Allowed range: 1 ns to 10000000 ns, at 1 ns steps.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_OUT\_LIMIT\_HIGH The value set is higher than max value.  
 RESULT\_DATA\_OUT\_LIMIT\_LOW The value set is lower than min value.

## See also

[PDMIR\\_GetTon\(\)](#)

#### 5.3.2.12 DIISDKExport RESULT PDMIR\_SetTTLout ( UINT16 *serialNumber*, OUTMODE *TTLout* )

Set the TTL output signal of the PDM-IR.

## Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>TTLOut</i>	output signal as OUTMODE typedef.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_OUT\_LIMIT\_HIGH The value set is higher than max value.  
 RESULT\_DATA\_OUT\_LIMIT\_LOW The value set is lower than min value.

## See also

[PDMIR\\_GetTTLOut\(\)](#)

#### 5.3.2.13 DIISDKExport RESULT PDMIR.SetNIMOut ( UINT16 serialNumber, OUTMODE NIMOut )

Set the NIM output signal of the PDM-IR.

## Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>NIMOut</i>	output signal as OUTMODE typedef.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_OUT\_LIMIT\_HIGH The value set is higher than max value.  
 RESULT\_DATA\_OUT\_LIMIT\_LOW The value set is lower than min value.

## See also

[PDMIR\\_GetNIMOut\(\)](#)

#### 5.3.2.14 DIISDKExport RESULT PDMIR.SetDelayTriggerIn ( UINT16 serialNumber, UINT8 delay )

Set the Trigger In delay of the PDM-IR.

## Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>delay</i>	Delay value in ns. Allowed range: 1 ns to 100 ns, at 1 ns steps.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_OUT\_LIMIT\_HIGH The value set is higher than max value.  
 RESULT\_DATA\_OUT\_LIMIT\_LOW The value set is lower than min value.

## See also

[PDMIR\\_GetDelayTriggerIn\(\)](#)

### 5.3.2.15 DIISDKExport RESULT PDMIR\_SetDelayTriggerInternal ( UINT16 *serialNumber*, UINT8 *delay* )

Set the Internal Trigger delay of the PDM-IR.

#### Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>delay</i>	Delay value in ns. Allowed range: 1 ns to 100 ns, at 1 ns steps.

#### Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_OUT\_LIMIT\_HIGH The value set is higher than max value.  
 RESULT\_DATA\_OUT\_LIMIT\_LOW The value set is lower than min value.

#### See also

[PDMIR\\_GetDelayTriggerInternal\(\)](#)

### 5.3.2.16 DIISDKExport RESULT PDMIR\_SetDelayAuxIn ( UINT16 *serialNumber*, UINT8 *delay* )

Set the Aux In delay of the PDM-IR.

#### Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>delay</i>	Delay value in ns. Allowed range: 1 ns to 100 ns, at 1 ns steps.

#### Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_OUT\_LIMIT\_HIGH The value set is higher than max value.  
 RESULT\_DATA\_OUT\_LIMIT\_LOW The value set is lower than min value.

#### See also

[PDMIR\\_GetDelayAuxIn\(\)](#)

### 5.3.2.17 DIISDKExport RESULT PDMIR\_SetDelayGate ( UINT16 *serialNumber*, UINT8 *delay* )

Set the Gate delay of the PDM-IR.

#### Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>delay</i>	Delay value in ns. Allowed range: 1 ns to 100 ns, at 1 ns steps.

#### Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_OUT\_LIMIT\_HIGH The value set is higher than max value.  
 RESULT\_DATA\_OUT\_LIMIT\_LOW The value set is lower than min value.

See also

[PDMIR\\_GetDelayGate\(\)](#)

#### 5.3.2.18 DIISDKExport RESULT PDMIR\_SetDelayNIMOut ( UINT16 *serialNumber*, UINT8 *delay* )

Set the NIM out delay of the PDM-IR.

##### Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>delay</i>	Delay value in ns. Allowed range: 1 ns to 100 ns, at 1 ns steps.

##### Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_OUT\_LIMIT\_HIGH The value set is higher than max value.  
 RESULT\_DATA\_OUT\_LIMIT\_LOW The value set is lower than min value.

See also

[PDMIR\\_GetDelayNIMOut\(\)](#)

#### 5.3.2.19 DIISDKExport RESULT PDMIR\_SetDelayTTLOut ( UINT16 *serialNumber*, UINT8 *delay* )

Set the TTL out delay of the PDM-IR.

##### Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>delay</i>	Delay value in ns. Allowed range: 1 ns to 100 ns, at 1 ns steps.

##### Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_OUT\_LIMIT\_HIGH The value set is higher than max value.  
 RESULT\_DATA\_OUT\_LIMIT\_LOW The value set is lower than min value.

See also

[PDMIR\\_GetDelayTTLOut\(\)](#)

#### 5.3.2.20 DIISDKExport RESULT PDMIR\_SetCounterIntegrationTime ( UINT16 *serialNumber*, UINT16 *integrationTime* )

Set the counters integration time of the PDM-IR.

##### Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>integrationTime</i>	Integration time in ms. Allowed range: 100 ms to 60000 ms, at 20 ms steps.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_OUT\_LIMIT\_HIGH The value set is higher than max value.  
 RESULT\_DATA\_OUT\_LIMIT\_LOW The value set is lower than min value.

## See also

[PDMIR\\_GetCounterIntegrationTime\(\)](#)

#### 5.3.2.21 DIISDKExport RESULT PDMIR\_SetCounterAvalancheEnable ( UINT16 *serialNumber*, COUNTERSTATUS *Cstatus* )

Set the photon out counter of the PDM-IR.

## Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>Cstatus</i>	counter status as COUNTERSTATUS typedef.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_OUT\_LIMIT\_HIGH The value set is higher than max value.  
 RESULT\_DATA\_OUT\_LIMIT\_LOW The value set is lower than min value.

## See also

[PDMIR\\_GetCounterAvalancheStatus\(\)](#)

#### 5.3.2.22 DIISDKExport RESULT PDMIR\_SetCounterValidGateEnable ( UINT16 *serialNumber*, COUNTERSTATUS *Cstatus* )

Set the valid gate counter of the PDM-IR.

## Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>Cstatus</i>	counter status as COUNTERSTATUS typedef.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_OUT\_LIMIT\_HIGH The value set is higher than max value.  
 RESULT\_DATA\_OUT\_LIMIT\_LOW The value set is lower than min value.

## See also

[PDMIR\\_GetCounterValidGateStatus\(\)](#)

#### 5.3.2.23 DIISDKExport RESULT PDMIR\_SetCounterAuxInEnable ( UINT16 *serialNumber*, COUNTERSTATUS *Cstatus* )

Set the Aux In counter of the PDM-IR.

## Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>Cstatus</i>	counter status as COUNTERSTATUS typedef.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_OUT\_LIMIT\_HIGH The value set is higher than max value.  
 RESULT\_DATA\_OUT\_LIMIT\_LOW The value set is lower than min value.

## See also

[PDMIR\\_GetCounterAuxInStatus\(\)](#)

#### 5.3.2.24 DIISDKExport RESULT PDMIR.SetCounterTriggerInEnable ( UINT16 serialNumber, COUNTERSTATUS Cstatus )

Set the trigger In counter of the PDM-IR.

## Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>Cstatus</i>	counter status as COUNTERSTATUS typedef.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_OUT\_LIMIT\_HIGH The value set is higher than max value.  
 RESULT\_DATA\_OUT\_LIMIT\_LOW The value set is lower than min value.

## See also

[PDMIR\\_GetCounterTriggerInStatus\(\)](#)

#### 5.3.2.25 DIISDKExport RESULT PDMIR.SetCounterInternalTriggerEnable ( UINT16 serialNumber, COUNTERSTATUS Cstatus )

Set the internal trigger counter of the PDM-IR.

## Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>Cstatus</i>	counter status as COUNTERSTATUS typedef.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_OUT\_LIMIT\_HIGH The value set is higher than max value.  
 RESULT\_DATA\_OUT\_LIMIT\_LOW The value set is lower than min value.

See also

[PDMIR\\_GetCounterInternalTriggerStatus\(\)](#)

#### 5.3.2.26 DIISDKExport RESULT PDMIR\_SetModuleOnOff ( UINT16 *serialNumber*, STATUSBITS *Mstatus* )

Set the module status of the PDM-IR.

##### Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>Mstatus</i>	counter status as STATUSBITS typedef. Allowed values: STATUSBITS_OFF, STATUSBITS_ON

##### Returns

RESULT\_OK

RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.

RESULT\_DATA\_OUT\_LIMIT\_HIGH The value set is higher than max value.

RESULT\_DATA\_OUT\_LIMIT\_LOW The value set is lower than min value.



## 5.4 Get methods

### Functions

- DIISDKExport [RESULT PDMIR\\_GetAmplitude](#) (UINT16 serialNumber, [UINT16](#) \*ampi)
- DIISDKExport [RESULT PDMIR\\_GetTemperature](#) (UINT16 serialNumber, [TEMP\\_VALUES](#) \*temp)
- DIISDKExport [RESULT PDMIR\\_GetHoldOff](#) (UINT16 serialNumber, [UINT32](#) \*holdOff, [HOLDOFFTYPE](#) \*type)
- DIISDKExport [RESULT PDMIR\\_GetGateMode](#) (UINT16 serialNumber, [GATEMODE](#) \*gate)
- DIISDKExport [RESULT PDMIR\\_GetGateShape](#) (UINT16 serialNumber, [GATESHAPE](#) \*shape)
- DIISDKExport [RESULT PDMIR\\_GetTriggerEdge](#) (UINT16 serialNumber, [TREDGE](#) \*edge)
- DIISDKExport [RESULT PDMIR\\_GetTriggerInTh](#) (UINT16 serialNumber, [INT16](#) \*TriggerInTh)
- DIISDKExport [RESULT PDMIR\\_GetAuxInTh](#) (UINT16 serialNumber, [INT16](#) \*AuxInTh)
- DIISDKExport [RESULT PDMIR\\_GetTriggerFunction](#) (UINT16 serialNumber, [TRFUNCT](#) \*function)
- DIISDKExport [RESULT PDMIR\\_GetFrequency](#) (UINT16 serialNumber, [UINT32](#) \*freq)
- DIISDKExport [RESULT PDMIR\\_GetTon](#) (UINT16 serialNumber, [UINT32](#) \*Ton)
- DIISDKExport [RESULT PDMIR\\_GetTTLOut](#) (UINT16 serialNumber, [OUTMODE](#) \*TTLOut)
- DIISDKExport [RESULT PDMIR\\_GetNIMOut](#) (UINT16 serialNumber, [OUTMODE](#) \*NIMOut)
- DIISDKExport [RESULT PDMIR\\_GetDelayTriggerIn](#) (UINT16 serialNumber, [UINT8](#) \*delay)
- DIISDKExport [RESULT PDMIR\\_GetDelayTriggerInternal](#) (UINT16 serialNumber, [UINT8](#) \*delay)
- DIISDKExport [RESULT PDMIR\\_GetDelayAuxIn](#) (UINT16 serialNumber, [UINT8](#) \*delay)
- DIISDKExport [RESULT PDMIR\\_GetDelayGate](#) (UINT16 serialNumber, [UINT8](#) \*delay)
- DIISDKExport [RESULT PDMIR\\_GetDelayNIMOut](#) (UINT16 serialNumber, [UINT8](#) \*delay)
- DIISDKExport [RESULT PDMIR\\_GetDelayTTLOut](#) (UINT16 serialNumber, [UINT8](#) \*delay)
- DIISDKExport [RESULT PDMIR\\_GetCounterIntegrationTime](#) (UINT16 serialNumber, [UINT16](#) \*integrationTime)
- DIISDKExport [RESULT PDMIR\\_GetCounterAvalancheStatus](#) (UINT16 serialNumber, [COUNTERSTATUS](#) \*status)
- DIISDKExport [RESULT PDMIR\\_GetCounterValidGateStatus](#) (UINT16 serialNumber, [COUNTERSTATUS](#) \*status)
- DIISDKExport [RESULT PDMIR\\_GetCounterAuxInStatus](#) (UINT16 serialNumber, [COUNTERSTATUS](#) \*status)
- DIISDKExport [RESULT PDMIR\\_GetCounterTriggerInStatus](#) (UINT16 serialNumber, [COUNTERSTATUS](#) \*status)
- DIISDKExport [RESULT PDMIR\\_GetCounterInternalTriggerStatus](#) (UINT16 serialNumber, [COUNTERSTATUS](#) \*status)
- DIISDKExport [RESULT PDMIR\\_GetAllCountersStatus](#) (UINT16 serialNumber, [COUNTERSTATUS](#) \*avalancheStatus, [COUNTERSTATUS](#) \*validGateStatus, [COUNTERSTATUS](#) \*auxInStatus, [COUNTERSTATUS](#) \*triggerInStatus, [COUNTERSTATUS](#) \*internalTriggerStatus)
- DIISDKExport [RESULT PDMIR\\_GetCounterAvalancheLastValue](#) (UINT16 serialNumber, [UINT8](#) \*index, [UINT32](#) \*counts)
- DIISDKExport [RESULT PDMIR\\_GetCounterValidGateLastValue](#) (UINT16 serialNumber, [UINT8](#) \*index, [UINT32](#) \*counts)
- DIISDKExport [RESULT PDMIR\\_GetCounterAuxInLastValue](#) (UINT16 serialNumber, [UINT8](#) \*index, [UINT32](#) \*counts)
- DIISDKExport [RESULT PDMIR\\_GetCounterTriggerInLastValue](#) (UINT16 serialNumber, [UINT8](#) \*index, [UINT32](#) \*counts)
- DIISDKExport [RESULT PDMIR\\_GetCounterInternalTriggerLastValue](#) (UINT16 serialNumber, [UINT8](#) \*index, [UINT32](#) \*counts)
- DIISDKExport [RESULT PDMIR\\_GetAllCountersLastValue](#) (UINT16 serialNumber, [UINT8](#) \*index, [UINT32](#) \*avalancheCounts, [UINT32](#) \*validGateCounts, [UINT32](#) \*auxInCounts, [UINT32](#) \*triggerInCounts, [UINT32](#) \*internalTriggerCounts)
- DIISDKExport [RESULT PDMIR\\_GetCounterAvalancheValueAtIndex](#) (UINT16 serialNumber, [UINT8](#) \*index, [UINT32](#) \*counts)
- DIISDKExport [RESULT PDMIR\\_GetCounterValidGateValueAtIndex](#) (UINT16 serialNumber, [UINT8](#) \*index, [UINT32](#) \*counts)
- DIISDKExport [RESULT PDMIR\\_GetCounterAuxInValueAtIndex](#) (UINT16 serialNumber, [UINT8](#) \*index, [UINT32](#) \*counts)

- DIISDKExport [RESULT PDMIR\\_GetCounterTriggerInValueAtIndex](#) (UINT16 serialNumber, UINT8 \*index, UINT32 \*counts)
- DIISDKExport [RESULT PDMIR\\_GetCounterInternalTriggerValueAtIndex](#) (UINT16 serialNumber, UINT8 \*index, UINT32 \*counts)
- DIISDKExport [RESULT PDMIR\\_GetAllCountersValueAtIndex](#) (UINT16 serialNumber, UINT8 \*index, UINT32 \*avalancheCounts, UINT32 \*validGateCounts, UINT32 \*auxInCounts, UINT32 \*triggerInCounts, UINT32 \*internalTriggerCounts)
- DIISDKExport [RESULT PDMIR\\_GetModuleInfo](#) (UINT16 serialNumber, MODULEINFO \*info)
- DIISDKExport [RESULT PDMIR\\_GetModuleStatus](#) (UINT16 serialNumber, STATUSBITS \*mStatus, STATUSBITS \*mTemperature, STATUSBITS \*mSpad, STATUSBITS \*mGate, STATUSWARMUP \*mWarm, UINT32 \*errors)
- DIISDKExport void [PDMIR\\_ErrorTranslator](#) (UINT32 error, char \*stringOut)

### 5.4.1 Detailed Description

Functions to get parameters of the PDM-IR.

### 5.4.2 Function Documentation

#### 5.4.2.1 DIISDKExport [RESULT PDMIR.GetAmplitude](#) ( [UINT16 serialNumber](#), [UINT16 \\* ampi](#) )

Get the gate amplitude of the PDM-IR.

##### Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>ampi</i>	Pointer to actual Amplitude in mV. This parameter is referenced.

##### Returns

[RESULT\\_OK](#)  
[RESULT\\_ERROR\\_COMMUNICATION](#) Error communicating with the PDM-IR.  
[RESULT\\_DATA\\_ERROR](#) The received value has wrong format.

##### See also

[PDMIR\\_GetGateShape\(\)](#)  
[PDMIR\\_SetAmplitude\(\)](#)

#### 5.4.2.2 DIISDKExport [RESULT PDMIR.GetTemperature](#) ( [UINT16 serialNumber](#), [TEMP\\_VALUES \\* temp](#) )

Get the working temperature of the PDM-IR.

##### Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>temp</i>	Pointer to actual Temperature as TEMP_VALUES. This parameter is referenced.

##### Returns

[RESULT\\_OK](#)  
[RESULT\\_ERROR\\_COMMUNICATION](#) Error communicating with the PDM-IR.  
[RESULT\\_DATA\\_ERROR](#) The received value has wrong format.

See also

[PDMIR\\_SetTemperature\(\)](#)

#### 5.4.2.3 DIISDKExport RESULT PDMIR.GetHoldOff ( UINT16 *serialNumber*, UINT32 \* *holdOff*, HOLDOFFTYPE \* *type* )

Get the hold off time of the PDM-IR.

Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>holdOff</i>	Pointer to actual Hold Off time in ns. This parameter is referenced.
<i>type</i>	Pointer to the Hold Off type. This parameter is referenced.

Returns

RESULT\_OK

RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.

RESULT\_DATA\_ERROR The received value has wrong format.

See also

[PDMIR\\_SetHoldOff\(\)](#)

#### 5.4.2.4 DIISDKExport RESULT PDMIR.GetGateMode ( UINT16 *serialNumber*, GATEMODE \* *gate* )

Get the gate mode of the PDM-IR.

Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>gate</i>	Pointer to gate mode as GATEMODE typedef. This parameter is referenced.

Returns

RESULT\_OK

RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.

RESULT\_DATA\_ERROR The received value has wrong format.

See also

[PDMIR\\_SetGateMode\(\)](#)

#### 5.4.2.5 DIISDKExport RESULT PDMIR.GetGateShape ( UINT16 *serialNumber*, GATESHAPE \* *shape* )

Get the gate shape of the PDM-IR.

Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>shape</i>	Pointer to gate shape as GATESHAPE typedef. This parameter is referenced.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_ERROR The received value has wrong format.

## See also

[PDMIR\\_SetGateShape\(\)](#)

#### 5.4.2.6 DIISDKExport RESULT PDMIR.GetTriggerEdge ( UINT16 *serialNumber*, TREDGE \* *edge* )

Get the Trigger Edge of the PDM-IR.

## Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>edge</i>	Pointer to the trigger edge as TREDGE typedef. This parameter is referenced.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_ERROR The received value has wrong format.

## See also

[PDMIR\\_SetTriggerEdge\(\)](#)

#### 5.4.2.7 DIISDKExport RESULT PDMIR.GetTriggerInTh ( UINT16 *serialNumber*, INT16 \* *TriggerInTh* )

Get the Trigger In Threshold of the PDM-IR.

## Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>TriggerInTh</i>	Pointer to the trigger In Threshold in mV. This parameter is referenced.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_ERROR The received value has wrong format.

## See also

[PDMIR\\_SetTriggerInTh\(\)](#)

#### 5.4.2.8 DIISDKExport RESULT PDMIR.GetAuxInTh ( UINT16 *serialNumber*, INT16 \* *AuxInTh* )

Get the Aux In Threshold of the PDM-IR.

## Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>AuxInTh</i>	Pointer to the Aux In Threshold in mV. This parameter is referenced.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_ERROR The received value has wrong format.

## See also

[PDMIR\\_SetAuxInTh\(\)](#)

## 5.4.2.9 DIISDKExport RESULT PDMIR.GetTriggerFunction ( UINT16 serialNumber, TRFUNCT \* function )

Get the Function between Aux In and trigger (in or internal) of the PDM-IR.

## Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>function</i>	Pointer to trigger function as TRFUNCT typedef. This parameter is referenced.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_ERROR The received value has wrong format.

## See also

[PDMIR\\_SetTriggerFunction\(\)](#)

## 5.4.2.10 DIISDKExport RESULT PDMIR.GetFrequency ( UINT16 serialNumber, UINT32 \* freq )

Get the internal trigger frequency in Hz of the PDM-IR.

## Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>freq</i>	Pointer to Internal trigger frequency in Hz. This parameter is referenced.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_ERROR The received value has wrong format.

## See also

[PDMIR\\_SetFrequency\(\)](#)

## 5.4.2.11 DIISDKExport RESULT PDMIR.GetTon ( UINT16 serialNumber, UINT32 \* Ton )

Get the Ton time in ns of the PDM-IR.

## Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>Ton</i>	Pointer to Ton time in ns. This parameter is referenced.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_ERROR The received value has wrong format.

## See also

[PDMIR\\_SetTon\(\)](#)

#### 5.4.2.12 DIISDKExport RESULT PDMIR\_GetTTLOut ( UINT16 *serialNumber*, OUTMODE \* *TTLOut* )

Get the TTL output signal of the PDM-IR.

## Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>TTLOut</i>	Pointer to output signal as OUTMODE typedef. This parameter is referenced.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_ERROR The received value has wrong format.

## See also

[PDMIR\\_SetTTLOut\(\)](#)

#### 5.4.2.13 DIISDKExport RESULT PDMIR\_GetNIMOut ( UINT16 *serialNumber*, OUTMODE \* *NIMOut* )

Get the NIM output signal of the PDM-IR.

## Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>NIMOut</i>	Pointer to output signal as OUTMODE typedef. This parameter is referenced.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_ERROR The received value has wrong format.

## See also

[PDMIR\\_SetNIMOut\(\)](#)

#### 5.4.2.14 DIISDKExport RESULT PDMIR\_GetDelayTriggerIn ( UINT16 *serialNumber*, UINT8 \* *delay* )

Get the Trigger In delay of the PDM-IR.

## Parameters



<i>serialNumber</i>	PDM-IR serial number
<i>delay</i>	Pointer to added delay at the signal in ns. This parameter is referenced.

**Returns**

RESULT\_OK

RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.

RESULT\_DATA\_ERROR The received value has wrong format.

**See also**[PDMIR\\_SetDelayTriggerIn\(\)](#)**5.4.2.15 DIISDKExport RESULT PDMIR\_GetDelayTriggerInternal ( UINT16 serialNumber, UINT8 \* delay )**

Get the Internal Trigger delay of the PDM-IR.

**Parameters**

<i>serialNumber</i>	PDM-IR serial number
<i>delay</i>	Pointer to added delay at the signal in ns. This parameter is referenced.

**Returns**

RESULT\_OK

RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.

RESULT\_DATA\_ERROR The received value has wrong format.

**See also**[PDMIR\\_SetDelayTriggerInternal\(\)](#)**5.4.2.16 DIISDKExport RESULT PDMIR\_GetDelayAuxIn ( UINT16 serialNumber, UINT8 \* delay )**

Get the Aux In delay of the PDM-IR.

**Parameters**

<i>serialNumber</i>	PDM-IR serial number
<i>delay</i>	Pointer to added delay at the signal in ns. This parameter is referenced.

**Returns**

RESULT\_OK

RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.

RESULT\_DATA\_ERROR The received value has wrong format.

**See also**[PDMIR\\_SetDelayAuxIn\(\)](#)

**5.4.2.17** DIISDKExport **RESULT** PDMIR\_GetDelayGate ( **UINT16** *serialNumber*, **UINT8** \* *delay* )

Get the Gate delay of the PDM-IR.

**Parameters**

<i>serialNumber</i>	PDM-IR serial number
<i>delay</i>	Pointer to added delay at the signal in ns. This parameter is referenced.

**Returns**

RESULT\_OK

RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.

RESULT\_DATA\_ERROR The received value has wrong format.

**See also**

[PDMIR\\_SetDelayGate\(\)](#)

**5.4.2.18** DIISDKExport **RESULT** PDMIR\_GetDelayNIMOut ( **UINT16** *serialNumber*, **UINT8** \* *delay* )

Get the NIM out delay of the PDM-IR.

**Parameters**

<i>serialNumber</i>	PDM-IR serial number
<i>delay</i>	Pointer to added delay at the signal in ns. This parameter is referenced.

**Returns**

RESULT\_OK

RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.

RESULT\_DATA\_ERROR The received value has wrong format.

**See also**

[PDMIR\\_SetDelayNIMOut\(\)](#)

**5.4.2.19** DIISDKExport **RESULT** PDMIR\_GetDelayTTLOut ( **UINT16** *serialNumber*, **UINT8** \* *delay* )

Get the TTL out delay of the PDM-IR.

**Parameters**

<i>serialNumber</i>	PDM-IR serial number
<i>delay</i>	Pointer to added delay at the signal in ns. This parameter is referenced.

**Returns**

RESULT\_OK

RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.

RESULT\_DATA\_ERROR The received value has wrong format.



See also

[PDMIR\\_SetDelayTTLOut\(\)](#)

#### 5.4.2.20 DIISDKExport RESULT PDMIR\_GetCounterIntegrationTime ( UINT16 *serialNumber*, UINT16 \* *integrationTime* )

Get the counters integration time of the PDM-IR.

##### Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>integrationTime</i>	Pointer to the integration time in ms. This parameter is referenced.

##### Returns

RESULT\_OK

RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.

RESULT\_DATA\_ERROR The received value has wrong format.

See also

[PDMIR\\_SetCounterIntegrationTime\(\)](#)

#### 5.4.2.21 DIISDKExport RESULT PDMIR\_GetCounterAvalancheStatus ( UINT16 *serialNumber*, COUNTERSTATUS \* *status* )

Get the photon out counter of the PDM-IR.

##### Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>status</i>	Pointer to the counter status as COUNTERSTATUS typedef. This parameter is referenced.

##### Returns

RESULT\_OK

RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.

RESULT\_DATA\_ERROR The received value has wrong format.

See also

[PDMIR\\_SetCounterAvalancheEnable\(\)](#)

#### 5.4.2.22 DIISDKExport RESULT PDMIR\_GetCounterValidGateStatus ( UINT16 *serialNumber*, COUNTERSTATUS \* *status* )

Get the valid gate counter of the PDM-IR.

##### Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>status</i>	Pointer to the counter status as COUNTERSTATUS typedef. This parameter is referenced.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_ERROR The received value has wrong format.

## See also

[PDMIR\\_SetCounterValidGateEnable\(\)](#)

#### 5.4.2.23 DIISDKExport RESULT PDMIR\_GetCounterAuxInStatus ( UINT16 serialNumber, COUNTERSTATUS \* status )

Get the Aux In counter of the PDM-IR.

## Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>status</i>	Pointer to the counter status as COUNTERSTATUS typedef. This parameter is referenced.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_ERROR The received value has wrong format.

## See also

[PDMIR\\_SetCounterAuxInEnable\(\)](#)

#### 5.4.2.24 DIISDKExport RESULT PDMIR\_GetCounterTriggerInStatus ( UINT16 serialNumber, COUNTERSTATUS \* status )

Get the trigger In counter of the PDM-IR.

## Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>status</i>	Pointer to the counter status as COUNTERSTATUS typedef. This parameter is referenced.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_ERROR The received value has wrong format.

## See also

[PDMIR\\_SetCounterTriggerInEnable\(\)](#)

#### 5.4.2.25 DIISDKExport RESULT PDMIR\_GetCounterInternalTriggerStatus ( UINT16 serialNumber, COUNTERSTATUS \* status )

Get the internal trigger counter of the PDM-IR.

## Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>status</i>	Pointer to the counter status as COUNTERSTATUS typedef. This parameter is referenced.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_ERROR The received value has wrong format.

## See also

[PDMIR\\_SetCounterInternalTriggerEnable\(\)](#)

5.4.2.26 DIISDKExport RESULT PDMIR\_GetAllCountersStatus ( UINT16 *serialNumber*, COUNTERSTATUS  
 \* *avalancheStatus*, COUNTERSTATUS \* *validGateStatus*, COUNTERSTATUS \* *auxInStatus*,  
 COUNTERSTATUS \* *triggerInStatus*, COUNTERSTATUS \* *internalTriggerStatus* )

Get the status of all the counter of the PDM-IR.

## Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>avalancheStatus</i>	Pointer to the avalanche counter status as COUNTERSTATUS typedef. This parameter is referenced.
<i>validGateStatus</i>	Pointer to the valid gate counter status as COUNTERSTATUS typedef. This parameter is referenced.
<i>auxInStatus</i>	Pointer to the aux in counter status as COUNTERSTATUS typedef. This parameter is referenced.
<i>triggerInStatus</i>	Pointer to the trigger in counter status as COUNTERSTATUS typedef. This parameter is referenced.
<i>internalTrigger-Status</i>	Pointer to the internal trigger counter status as COUNTERSTATUS typedef. This parameter is referenced.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_ERROR The received value has wrong format.

5.4.2.27 DIISDKExport RESULT PDMIR\_GetCounterAvalancheLastValue ( UINT16 *serialNumber*, UINT8 \* *index*, UINT32 \*  
*counts* )

Get the photon out counter value of the PDM-IR.

## Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>index</i>	Pointer to the index of the rotating array. Index is a value between 0 and 19. This parameter is referenced.
<i>counts</i>	Pointer to the counts value accumulated in the integration time. This parameter is referenced.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_ERROR The received value has wrong format.

## See also

[PDMIR\\_GetCounterAvalancheValueAtIndex\(\)](#)

5.4.2.28 **DIISDKExport RESULT PDMIR\_GetCounterValidGateLastValue ( UINT16 *serialNumber*, UINT8 \* *index*, UINT32 \* *counts* )**

Get the valid gate counter value of the PDM-IR.

## Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>index</i>	Pointer to the index of the rotating array. Index is a value between 0 and 19. This parameter is referenced.
<i>counts</i>	Pointer to the counts value accumulated in the integration time. This parameter is referenced.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_ERROR The received value has wrong format.

## See also

[PDMIR\\_GetCounterValidGateValueAtIndex\(\)](#)

5.4.2.29 **DIISDKExport RESULT PDMIR\_GetCounterAuxInLastValue ( UINT16 *serialNumber*, UINT8 \* *index*, UINT32 \* *counts* )**

Get the aux in counter value of the PDM-IR.

## Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>index</i>	Pointer to the index of the rotating array. Index is a value between 0 and 19. This parameter is referenced.
<i>counts</i>	Pointer to the counts value accumulated in the integration time. This parameter is referenced.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_ERROR The received value has wrong format.

## See also

[PDMIR\\_GetCounterAuxInValueAtIndex\(\)](#)

#### 5.4.2.30 DIISDKExport RESULT PDMIR\_GetCounterTriggerInLastValue ( UINT16 *serialNumber*, UINT8 \* *index*, UINT32 \* *counts* )

Get the trigger in counter value of the PDM-IR.

##### Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>index</i>	Pointer to the index of the rotating array. Index is a value between 0 and 19. This parameter is referenced.
<i>counts</i>	Pointer to the counts value accumulated in the integration time. This parameter is referenced.

##### Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_ERROR The received value has wrong format.

##### See also

[PDMIR\\_GetCounterTriggerInValueAtIndex\(\)](#)

#### 5.4.2.31 DIISDKExport RESULT PDMIR\_GetCounterInternalTriggerLastValue ( UINT16 *serialNumber*, UINT8 \* *index*, UINT32 \* *counts* )

Get the internal trigger counter value of the PDM-IR.

##### Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>index</i>	Pointer to the index of the rotating array. Index is a value between 0 and 19. This parameter is referenced.
<i>counts</i>	Pointer to the counts value accumulated in the integration time. This parameter is referenced.

##### Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_ERROR The received value has wrong format.

##### See also

[PDMIR\\_GetCounterInternalTriggerValueAtIndex\(\)](#)

#### 5.4.2.32 DIISDKExport RESULT PDMIR\_GetAllCountersLastValue ( UINT16 *serialNumber*, UINT8 \* *index*, UINT32 \* *avalancheCounts*, UINT32 \* *validGateCounts*, UINT32 \* *auxInCounts*, UINT32 \* *triggerInCounts*, UINT32 \* *internalTriggerCounts* )

Get the value of all the counters of the PDM-IR.

##### Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>index</i>	Pointer to the index of the rotating array. Index is a value between 0 and 19. This parameter is referenced.

<i>avalancheCounts</i>	Pointer to the counts value accumulated in the integration time in the avalanche counter. This parameter is referenced.
<i>validGateCounts</i>	Pointer to the counts value accumulated in the integration time in the valid gate counter. This parameter is referenced.
<i>auxInCounts</i>	Pointer to the counts value accumulated in the integration time in the aux in counter. This parameter is referenced.
<i>triggerInCounts</i>	Pointer to the counts value accumulated in the integration time in the trigger in counter. This parameter is referenced.
<i>internalTrigger-Counts</i>	Pointer to the counts value accumulated in the integration time in the internal trigger counter. This parameter is referenced.

**Returns**

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_ERROR The received value has wrong format.

**See also**

[PDMIR\\_GetAllCountersValueAtIndex\(\)](#)

#### 5.4.2.33 DIISDKExport RESULT PDMIR\_GetCounterAvalancheValueAtIndex ( UINT16 serialNumber, UINT8 \* index, UINT32 \* counts )

Get the photon out counter value of the PDM-IR at specified index.

**Parameters**

<i>serialNumber</i>	PDM-IR serial number
<i>index</i>	Pointer to the index of the rotating array. Index is a value between 0 and 19. This parameter is referenced.
<i>counts</i>	Pointer to the counts value accumulated int the integration time. This parameter is referenced.

**Returns**

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_OUT\_LIMIT\_HIGH The index value is higher than max value.  
 RESULT\_DATA\_OUT\_LIMIT\_LOW The index value is lower than min value.  
 RESULT\_DATA\_ERROR The received value has wrong format.


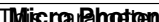
**See also**

[PDMIR\\_GetCounterAvalancheLastValue\(\)](#)

#### 5.4.2.34 DIISDKExport RESULT PDMIR\_GetCounterValidGateValueAtIndex ( UINT16 serialNumber, UINT8 \* index, UINT32 \* counts )

Get the valid gate counter value of the PDM-IR at specified index.

**Parameters**

<i>serialNumber</i>	PDM-IR serial number
<i>index</i>	Pointer to the index of the rotating array. Index is a value between 0 and 19. This parameter is referenced.
 <i>counts</i>	Pointer to the counts value accumulated int the integration time.  <b>Micro Photon Devices</b> <small>sd.</small>

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_OUT\_LIMIT\_HIGH The index value is higher than max value.  
 RESULT\_DATA\_OUT\_LIMIT\_LOW The index value is lower than min value.  
 RESULT\_DATA\_ERROR The received value has wrong format.

## See also

[PDMIR\\_GetCounterValidGateLastValue\(\)](#)

#### 5.4.2.35 DIISDKExport RESULT PDMIR\_GetCounterAuxInValueAtIndex ( UINT16 serialNumber, UINT8 \* index, UINT32 \* counts )

Get the Aux In counter value of the PDM-IR at specified index.

## Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>index</i>	Pointer to the index of the rotating array. Index is a value between 0 and 19. This parameter is referenced.
<i>counts</i>	Pointer to the counts value accumulated int the integration time. This parameter is referenced.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_OUT\_LIMIT\_HIGH The index value is higher than max value.  
 RESULT\_DATA\_OUT\_LIMIT\_LOW The index value is lower than min value.  
 RESULT\_DATA\_ERROR The received value has wrong format.

## See also

[PDMIR\\_GetCounterAuxInLastValue\(\)](#)

#### 5.4.2.36 DIISDKExport RESULT PDMIR\_GetCounterTriggerInValueAtIndex ( UINT16 serialNumber, UINT8 \* index, UINT32 \* counts )

Get the trigger in counter value of the PDM-IR at specified index.

## Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>index</i>	Pointer to the index of the rotating array. Index is a value between 0 and 19. This parameter is referenced.
<i>counts</i>	Pointer to the counts value accumulated int the integration time. This parameter is referenced.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_OUT\_LIMIT\_HIGH The index value is higher than max value.  
 RESULT\_DATA\_OUT\_LIMIT\_LOW The index value is lower than min value.  
 RESULT\_DATA\_ERROR The received value has wrong format.

See also

[PDMIR\\_GetCounterTriggerInLastValue\(\)](#)

5.4.2.37 DIISDKExport **RESULT** PDMIR\_GetCounterInternalTriggerValueAtIndex ( **UINT16** *serialNumber*, **UINT8** \* *index*, **UINT32** \* *counts* )

Get the internal trigger counter value of the PDM-IR at specified index.

#### Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>index</i>	Pointer to the index of the rotating array. Index is a value between 0 and 19. This parameter is referenced.
<i>counts</i>	Pointer to the counts value accumulated int the integration time. This parameter is referenced.

#### Returns

**RESULT\_OK**  
**RESULT\_ERROR\_COMMUNICATION** Error communicating with the PDM-IR.  
**RESULT\_DATA\_OUT\_LIMIT\_HIGH** The index value is higher than max value.  
**RESULT\_DATA\_OUT\_LIMIT\_LOW** The index value is lower than min value.  
**RESULT\_DATA\_ERROR** The received value has wrong format.

See also

[PDMIR\\_GetCounterInternalTriggerLastValue\(\)](#)

5.4.2.38 DIISDKExport **RESULT** PDMIR\_GetAllCountersValueAtIndex ( **UINT16** *serialNumber*, **UINT8** \* *index*, **UINT32** \* *avalancheCounts*, **UINT32** \* *validGateCounts*, **UINT32** \* *auxInCounts*, **UINT32** \* *triggerInCounts*, **UINT32** \* *internalTriggerCounts* )

Get the value of all the counters of the PDM-IR at specified index.

#### Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>index</i>	Pointer to the index of the rotating array. Index is a value between 0 and 19. This parameter is referenced.
<i>avalancheCounts</i>	Pointer to the counts value accumulated in the integration time in the avalanche counter. This parameter is referenced.
<i>validGateCounts</i>	Pointer to the counts value accumulated in the integration time in the valid gate counter. This parameter is referenced.
<i>auxInCounts</i>	Pointer to the counts value accumulated in the integration time in the aux in counter. This parameter is referenced.
<i>triggerInCounts</i>	Pointer to the counts value accumulated in the integration time in the trigger in counter. This parameter is referenced.
<i>internalTrigger-Counts</i>	Pointer to the counts value accumulated in the integration time in the internal trigger counter. This parameter is referenced.

#### Returns

**RESULT\_OK**  
**RESULT\_ERROR\_COMMUNICATION** Error communicating with the PDM-IR.  
**RESULT\_DATA\_OUT\_LIMIT\_HIGH** The index value is higher than max value.  
**RESULT\_DATA\_OUT\_LIMIT\_LOW** The index value is lower than min value.



RESULT\_DATA\_ERROR The received value has wrong format.

See also

[PDMIR\\_GetAllCountersLastValue\(\)](#)

#### 5.4.2.39 DIISDKExport RESULT PDMIR\_GetModuleInfo ( UINT16 *serialNumber*, MODULEINFO \* *info* )

Get the module information of the PDM-IR.

Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>info</i>	Pointer to the structure <a href="#">MODULEINFO</a> . This parameter is referenced.

Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_ERROR The received value has wrong format.

#### 5.4.2.40 DIISDKExport RESULT PDMIR\_GetModuleStatus ( UINT16 *serialNumber*, STATUSBITS \* *mStatus*, STATUSBITS \* *mTemperature*, STATUSBITS \* *mSpad*, STATUSBITS \* *mGate*, STATUSWARMUP \* *mWarm*, UINT32 \* *errors* )

Get the module status of the PDM-IR.

Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>mStatus</i>	Pointer to the module status. Allowed values: STATUSBITS_OFF, STATUSBITS_ON. This parameter is referenced.
<i>mTemperature</i>	Pointer to the temperature status. Allowed values: STATUSBITS_OFF, STATUSBITS_ON, STATUSBITS_Updating. This parameter is referenced.
<i>mSpad</i>	Pointer to the SPAD polarization status. Allowed values: STATUSBITS_OFF, STATUSBITS_ON, STATUSBITS_Updating. This parameter is referenced.
<i>mGate</i>	Pointer to the gate status. Allowed values: STATUSBITS_OFF, STATUSBITS_ON, STATUSBITS_Updating. This parameter is referenced.
<i>mWarm</i>	Pointer to the module temperature. This parameter is referenced.
<i>errors</i>	Pointer to error code. Use <a href="#">PDMIR_ErrorTranslator()</a> to convert code value to string. This parameter is referenced.

Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_ERROR The received value has wrong format.

See also

[PDMIR\\_ErrorTranslator\(\)](#)

#### 5.4.2.41 DIISDKExport void PDMIR\_ErrorTranslator ( UINT32 *error*, char \* *stringOut* )

Get the string description of an error code of the PDM-IR.

## Parameters

<i>error</i>	error code.
<i>stringOut</i>	Pointer to output char array. Array is at least 450 char lenght. This parameter is referenced.

## Returns

null

## See also

[PDMIR\\_GetModuleStatus\(\)](#)

## 5.5 Configuration methods

### Functions

- DIISDKExport [RESULT PDMIR\\_SaveCurrentConfig](#) (UINT16 serialNumber, UINT8 configNumber, char \*configName)
- DIISDKExport [RESULT PDMIR\\_SaveCurrentConfigForceOverWrite](#) (UINT16 serialNumber, UINT8 configNumber, char \*configName)
- DIISDKExport [RESULT PDMIR\\_DeleteConfig](#) (UINT16 serialNumber, UINT8 configNumber)
- DIISDKExport [RESULT PDMIR\\_SetConfigPowerUp](#) (UINT16 serialNumber, UINT8 configNumber)
- DIISDKExport [RESULT PDMIR\\_LoadConfig](#) (UINT16 serialNumber, UINT8 configNumber)
- DIISDKExport [RESULT PDMIR\\_GetCurrentConfig](#) (UINT16 serialNumber, MODULECONFIG \*config)
- DIISDKExport [RESULT PDMIR\\_GetConfigX](#) (UINT16 serialNumber, UINT8 configNumber, MODULECONFIG \*config, char \*configName)
- DIISDKExport [RESULT PDMIR\\_GetConfigPowerUp](#) (UINT16 serialNumber, UINT8 \*configNumber)

### 5.5.1 Detailed Description

Functions to manage the configurations of the PDM-IR.

### 5.5.2 Function Documentation

#### 5.5.2.1 DIISDKExport [RESULT PDMIR\\_SaveCurrentConfig](#) ( [UINT16 serialNumber](#), [UINT8 configNumber](#), char \* [configName](#) )

Store current configuration of the PDM-IR in the configNumber position.

#### Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>configNumber</i>	position where the configuration will be saved. Allowed values: from 4 to 10. If the position contains a valid configuration doesn't overwrite the configuration
<i>configName</i>	Pointer to a char array. Max length is 16 chars. The name needs to be terminated with 0.

#### Returns

[RESULT\\_OK](#)  
[RESULT\\_ERROR\\_COMMUNICATION](#) Error communicating with the PDM-IR or there's a valid configuration in the configuration position.  
[RESULT\\_DATA\\_OUT\\_LIMIT\\_HIGH](#) The position set is higher than max value.  
[RESULT\\_DATA\\_OUT\\_LIMIT\\_LOW](#) The position set is lower than min value.

#### See also

[PDMIR\\_LoadConfig\(\)](#)

#### 5.5.2.2 DIISDKExport [RESULT PDMIR\\_SaveCurrentConfigForceOverWrite](#) ( [UINT16 serialNumber](#), [UINT8 configNumber](#), char \* [configName](#) )

Store current configuration of the PDM-IR in the configNumber position overwriting the existing one.

## Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>configNumber</i>	position where save the configuration. Allowed values: from 4 to 10. If the position contains a valid configuration, it's overwritten
<i>configName</i>	Pointer to a char array. Max length is 16 chars. The name needs to be terminated with 0.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
 RESULT\_DATA\_OUT\_LIMIT\_HIGH The position set is higher than max value.  
 RESULT\_DATA\_OUT\_LIMIT\_LOW The position set is lower than min value.

## See also

[PDMIR\\_LoadConfig\(\)](#)

### 5.5.2.3 DIISDKExport RESULT PDMIR.DeleteConfig ( UINT16 serialNumber, UINT8 configNumber )

Delete the configuration stored in the configNumber position.

## Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>configNumber</i>	position where the configuration will be deleted. Allowed values: from 4 to 10.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR or the configuration doesn't exist.  
 RESULT\_DATA\_OUT\_LIMIT\_HIGH The position set is higher than max value.  
 RESULT\_DATA\_OUT\_LIMIT\_LOW The position set is lower than min value.

## See also

[PDMIR\\_LoadConfig\(\)](#)

### 5.5.2.4 DIISDKExport RESULT PDMIR.SetConfigPowerUp ( UINT16 serialNumber, UINT8 configNumber )

Set a stored configuration as power up configuration of the PDM-IR.

## Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>configNumber</i>	Configuration position set as power up configuration. Allowed values: from 1 to 10.

## Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR or the configuration in configNumber position doesn't exist.  
 RESULT\_DATA\_OUT\_LIMIT\_HIGH The position set is higher than max value.  
 RESULT\_DATA\_OUT\_LIMIT\_LOW The position set is lower than min value.

See also

[PDMIR\\_LoadConfig\(\)](#)

#### 5.5.2.5 DIISDKExport RESULT PDMIR.LoadConfig ( UINT16 serialNumber, UINT8 configNumber )

Load a stored configuration of the PDM-IR.

Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>configNumber</i>	position where the configuration is read and loaded. Allowed values: from 1 to 10.

Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR or the configuration doesn't exist.  
 RESULT\_DATA\_OUT\_LIMIT\_HIGH The position set is higher than max value.  
 RESULT\_DATA\_OUT\_LIMIT\_LOW The position set is lower than min value.

See also

[PDMIR\\_SaveCurrentConfig\(\)](#)  
[PDMIR\\_SaveCurrentConfigForceOverWrite\(\)](#)  
[PDMIR\\_DeleteConfig\(\)](#)  
[PDMIR\\_SetConfigPowerUp\(\)](#)

#### 5.5.2.6 DIISDKExport RESULT PDMIR.GetCurrentConfig ( UINT16 serialNumber, MODULECONFIG \* config )

Get the current configuration parameters of the PDM-IR.

Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>config</i>	Pointer to <a href="#">MODULECONFIG</a> structure.

Returns

RESULT\_OK  
 RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.

#### 5.5.2.7 DIISDKExport RESULT PDMIR.GetConfigX ( UINT16 serialNumber, UINT8 configNumber, MODULECONFIG \* config, char \* configName )

Get a stored configuration of the PDM-IR.

Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>configNumber</i>	position where the configuration is read. Allowed values: from 1 to 10.
<i>config</i>	Pointer to <a href="#">MODULECONFIG</a> structure. This parameter is referenced.
<i>configName</i>	Pointer to a char array containing the configuration name. Max length is 16 chars. This parameter is referenced.

## Returns

RESULT\_OK  
RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR.  
RESULT\_DATA\_OUT\_LIMIT\_HIGH The position set is higher than max value.  
RESULT\_DATA\_OUT\_LIMIT\_LOW The position set is lower than min value.

**5.5.2.8 DIISDKExport RESULT PDMIR.GetConfigPowerUp ( UINT16 *serialNumber*, UINT8 \* *configNumber* )**

Get the stored configuration number set as power up configuration of the PDM-IR.

## Parameters

<i>serialNumber</i>	PDM-IR serial number
<i>configNumber</i>	Pointer to power up configuration position. Allowed values: from 1 to 10. This parameter is referenced.

## Returns

RESULT\_OK  
RESULT\_ERROR\_COMMUNICATION Error communicating with the PDM-IR or the configuration doesn't exist.  
RESULT\_DATA\_OUT\_LIMIT\_HIGH The position set is higher than max value.  
RESULT\_DATA\_OUT\_LIMIT\_LOW The position set is lower than min value.

## Chapter 6

# Data Structure Documentation

### 6.1 MODULECONFIG Struct Reference

```
#include <PDM-IR_SDK.h>
```

#### Data Fields

- [UINT16 Amplitude](#)
- [TEMP\\_VALUES Temperature](#)
- [UINT32 HoldOff](#)
- [HOLDOFFTYPE HoldOffLevel](#)
- [GATEMODE GateMode](#)
- [GATESHAPE GateShape](#)
- [TREDGE TriggerEdge](#)
- [INT16 TriggerInTh](#)
- [INT16 AuxInTh](#)
- [TRFUNCT TriggerFunction](#)
- [UINT32 Frequency](#)
- [UINT32 Ton](#)
- [OUTMODE TtlOut](#)
- [OUTMODE NimOut](#)
- [UINT8 DelayTriggerIn](#)
- [UINT8 DelayTriggerInternal](#)
- [UINT8 DelayAuxIn](#)
- [UINT8 DelayGate](#)
- [UINT8 DelayNimOut](#)
- [UINT8 DelayTtlOut](#)
- [UINT16 CounterIntegrationTime](#)
- [COUNTERSTATUS CounterAvalancheStatus](#)
- [COUNTERSTATUS CounterValidGateStatus](#)
- [COUNTERSTATUS CounterAuxInStatus](#)
- [COUNTERSTATUS CounterTriggerInStatus](#)
- [COUNTERSTATUS CounterInternalTriggerStatus](#)

#### 6.1.1 Detailed Description

Module Configuration Structure.

It contains the configuration of the PDM-IR.

The documentation for this struct was generated from the following file:

- [PDM-IR\\_SDK.h](#)

## 6.2 MODULEINFO Struct Reference

```
#include <PDM-IR_SDK.h>
```

### Data Fields

- [UINT16](#) SN
- [SPAD\\_TYPE](#) SpadType
- char [SpadName](#) [16]
- char [FpgaFwVer](#) [16]
- char [FpgaHwVer](#) [16]
- char [McuFwVer](#) [16]
- char [McuHwVer](#) [16]

### 6.2.1 Detailed Description

Structure containing the information of the PDM-IR.

The documentation for this struct was generated from the following file:

- [PDM-IR\\_SDK.h](#)



## Chapter 7

# File Documentation

### 7.1 PDM-IR\_SDK.h File Reference

#### Data Structures

- struct [MODULECONFIG](#)
- struct [MODULEINFO](#)

#### Macros

- `#define` [WIN32\\_LEAN\\_AND\\_MEAN](#)

#### Typedefs

- typedef unsigned char [UINT8](#)
- typedef signed char [INT8](#)
- typedef signed short [INT16](#)
- typedef unsigned short [UINT16](#)
- typedef unsigned int [UINT32](#)
- typedef signed int [INT32](#)

#### Enumerations

- enum [RESULT](#) {  
    [RESULT\\_OK](#), [RESULT\\_DATA\\_OUT\\_LIMIT\\_HIGH](#), [RESULT\\_DATA\\_OUT\\_LIMIT\\_LOW](#), [RESULT\\_DATA\\_ERROR](#),  
    [RESULT\\_ERROR\\_COMMUNICATION](#) }
- enum [TEMP\\_VALUES](#) { [THIGH](#) = 0, [TMEDIUM](#) = 1, [TLOW](#) = 2, [TLOWEST](#) = 3 }
- enum [HOLDOFFTYPE](#) { [HOLDOFFTYPE\\_EDGE](#) = 0, [HOLDOFFTYPE\\_LEVEL](#) = 1 }
- enum [GATEMODE](#) { [GATEMODE\\_INTERNAL](#) = 0, [GATEMODE\\_EXTERNAL](#) = 1 }
- enum [GATESHAPE](#) { [GATESHAPE\\_FIXEDGATE](#) = 0, [GATESHAPE\\_FREEGATE](#) = 1, [GATESHAPE\\_FREERUNNING](#) = 2 }
- enum [TREDGE](#) { [TREDGE\\_LH](#) = 0, [TREDGE\\_HL](#) = 1 }
- enum [TRFUNCT](#) {  
    [In\\_AND\\_Aux](#) = 0, [In\\_OR\\_Aux](#) = 1, [In\\_XOR\\_Aux](#) = 2, [notIn\\_AND\\_Aux](#) = 3,  
    [notIn\\_OR\\_Aux](#) = 4, [notIn\\_XOR\\_Aux](#) = 5, [In\\_AND\\_notAux](#) = 6, [In\\_OR\\_notAux](#) = 7,  
    [In\\_XOR\\_notAux](#) = 8, [notIn\\_AND\\_notAux](#) = 9, [notIn\\_OR\\_notAux](#) = 10, [notIn\\_XOR\\_notAux](#) = 11,  
    [In\\_NAND\\_Aux](#) = 12, [In\\_NOR\\_Aux](#) = 13, [In\\_XNOR\\_Aux](#) = 14, [notIn\\_NAND\\_Aux](#) = 15,  
    [notIn\\_NOR\\_Aux](#) = 16, [notIn\\_XNOR\\_Aux](#) = 17, [In\\_NAND\\_notAux](#) = 18, [In\\_NOR\\_notAux](#) = 19,  
    [In\\_XNOR\\_notAux](#) = 20, [notIn\\_NAND\\_notAux](#) = 21, [notIn\\_NOR\\_notAux](#) = 22, [notIn\\_XNOR\\_notAux](#) = 23,  
    [only\\_notIn](#) = 24, [only\\_IN](#) = 25 }

- enum `OUTMODE` {  
`OUTMODE_TRIGGERGATE` = 0, `OUTMODE_PHOTONOUT` = 1, `OUTMODE_HOLDOFF` = 2, `OUTMODE_VALIDGATE` = 3,  
`OUTMODE_INTERNALTRIGGER` = 4, `OUTMODE_MODULESTATUS` = 5 }
- enum `COUNTERSTATUS` { `COUNTERSTATUS_OFF` = 0, `COUNTERSTATUS_ON` = 1, `COUNTERSTATUS_Continuous` = 2 }
- enum `SPAD_TYPE` {  
`TYPE_1` = 0x1, `TYPE_2` = 0x2, `TYPE_3` = 0x3, `TYPE_4` = 0x4,  
`TYPE_5` = 0x5, `TYPE_6` = 0x6, `TYPE_7` = 0x7 }
- enum `STATUSBITS` {  
`STATUSBITS_OFF` = 0, `STATUSBITS_ON` = 1, `STATUSBITS_Updating` = 2, `STATUSBITS_SafeValue` = 3,  
`STATUSBITS_ToBeUpdated` = 4, `STATUSBITS_Updated` = 5 }
- enum `STATUSWARMUP` { `STATUSWARMUP_isWarmingUp` = 0, `STATUSWARMUP_isSteadyState` = 1 }

## Functions

- `DIISDKExport RESULT PDMIR_SearchDevices` (`UINT16` \*arrayDataSN, `UINT16` \*numDevsFound)
- `DIISDKExport RESULT PDMIR_OpenCommunication` (`UINT16` serialNumber)
- `DIISDKExport RESULT PDMIR_CloseCommunication` (`UINT16` serialNumber)
- `DIISDKExport RESULT PDMIR_SetAmplitude` (`UINT16` serialNumber, `UINT16` AmplimV)
- `DIISDKExport RESULT PDMIR_SetTemperature` (`UINT16` serialNumber, `TEMP_VALUES` temp)
- `DIISDKExport RESULT PDMIR_SetHoldOff` (`UINT16` serialNumber, `UINT32` holdOff, `HOLDOFFTYPE` type)
- `DIISDKExport RESULT PDMIR_SetGateMode` (`UINT16` serialNumber, `GATEMODE` gm)
- `DIISDKExport RESULT PDMIR_SetGateShape` (`UINT16` serialNumber, `GATESHAPE` gs)
- `DIISDKExport RESULT PDMIR_SetTriggerEdge` (`UINT16` serialNumber, `TREDGE` TriggerEdge)
- `DIISDKExport RESULT PDMIR_SetTriggerInTh` (`UINT16` serialNumber, `INT16` TriggerTh)
- `DIISDKExport RESULT PDMIR_SetAuxInTh` (`UINT16` serialNumber, `INT16` TriggerTh)
- `DIISDKExport RESULT PDMIR_SetTriggerFunction` (`UINT16` serialNumber, `TRFUNCT` TriggerFunction)
- `DIISDKExport RESULT PDMIR_SetFrequency` (`UINT16` serialNumber, `UINT32` frequency)
- `DIISDKExport RESULT PDMIR_SetTon` (`UINT16` serialNumber, `UINT32` tOn)
- `DIISDKExport RESULT PDMIR_SetTTLOut` (`UINT16` serialNumber, `OUTMODE` TTLOut)
- `DIISDKExport RESULT PDMIR_SetNIMOut` (`UINT16` serialNumber, `OUTMODE` NIMOut)
- `DIISDKExport RESULT PDMIR_SetDelayTriggerIn` (`UINT16` serialNumber, `UINT8` delay)
- `DIISDKExport RESULT PDMIR_SetDelayTriggerInternal` (`UINT16` serialNumber, `UINT8` delay)
- `DIISDKExport RESULT PDMIR_SetDelayAuxIn` (`UINT16` serialNumber, `UINT8` delay)
- `DIISDKExport RESULT PDMIR_SetDelayGate` (`UINT16` serialNumber, `UINT8` delay)
- `DIISDKExport RESULT PDMIR_SetDelayNIMOut` (`UINT16` serialNumber, `UINT8` delay)
- `DIISDKExport RESULT PDMIR_SetDelayTTLOut` (`UINT16` serialNumber, `UINT8` delay)
- `DIISDKExport RESULT PDMIR_SetCounterIntegrationTime` (`UINT16` serialNumber, `UINT16` integrationTime)
- `DIISDKExport RESULT PDMIR_SetCounterAvalancheEnable` (`UINT16` serialNumber, `COUNTERSTATUS` Cstatus)
- `DIISDKExport RESULT PDMIR_SetCounterValidGateEnable` (`UINT16` serialNumber, `COUNTERSTATUS` Cstatus)
- `DIISDKExport RESULT PDMIR_SetCounterAuxInEnable` (`UINT16` serialNumber, `COUNTERSTATUS` Cstatus)
- `DIISDKExport RESULT PDMIR_SetCounterTriggerInEnable` (`UINT16` serialNumber, `COUNTERSTATUS` Cstatus)
- `DIISDKExport RESULT PDMIR_SetCounterInternalTriggerEnable` (`UINT16` serialNumber, `COUNTERSTATUS` Cstatus)
- `DIISDKExport RESULT PDMIR_SetModuleOnOff` (`UINT16` serialNumber, `STATUSBITS` Mstatus)
- `DIISDKExport RESULT PDMIR_GetAmplitude` (`UINT16` serialNumber, `UINT16` \*ampi)
- `DIISDKExport RESULT PDMIR_GetTemperature` (`UINT16` serialNumber, `TEMP_VALUES` \*temp)
- `DIISDKExport RESULT PDMIR_GetHoldOff` (`UINT16` serialNumber, `UINT32` \*holdOff, `HOLDOFFTYPE` \*type)
- `DIISDKExport RESULT PDMIR_GetGateMode` (`UINT16` serialNumber, `GATEMODE` \*gate)

- DIISDKExport [RESULT PDMIR\\_GetGateShape](#) (UINT16 serialNumber, [GATESHAPE](#) \*shape)
- DIISDKExport [RESULT PDMIR\\_GetTriggerEdge](#) (UINT16 serialNumber, [TREDGE](#) \*edge)
- DIISDKExport [RESULT PDMIR\\_GetTriggerInTh](#) (UINT16 serialNumber, [INT16](#) \*TriggerInTh)
- DIISDKExport [RESULT PDMIR\\_GetAuxInTh](#) (UINT16 serialNumber, [INT16](#) \*AuxInTh)
- DIISDKExport [RESULT PDMIR\\_GetTriggerFunction](#) (UINT16 serialNumber, [TRFUNCT](#) \*function)
- DIISDKExport [RESULT PDMIR\\_GetFrequency](#) (UINT16 serialNumber, [UINT32](#) \*freq)
- DIISDKExport [RESULT PDMIR\\_GetTon](#) (UINT16 serialNumber, [UINT32](#) \*Ton)
- DIISDKExport [RESULT PDMIR\\_GetTTLOut](#) (UINT16 serialNumber, [OUTMODE](#) \*TTLOut)
- DIISDKExport [RESULT PDMIR\\_GetNIMOut](#) (UINT16 serialNumber, [OUTMODE](#) \*NIMOut)
- DIISDKExport [RESULT PDMIR\\_GetDelayTriggerIn](#) (UINT16 serialNumber, [UINT8](#) \*delay)
- DIISDKExport [RESULT PDMIR\\_GetDelayTriggerInternal](#) (UINT16 serialNumber, [UINT8](#) \*delay)
- DIISDKExport [RESULT PDMIR\\_GetDelayAuxIn](#) (UINT16 serialNumber, [UINT8](#) \*delay)
- DIISDKExport [RESULT PDMIR\\_GetDelayGate](#) (UINT16 serialNumber, [UINT8](#) \*delay)
- DIISDKExport [RESULT PDMIR\\_GetDelayNIMOut](#) (UINT16 serialNumber, [UINT8](#) \*delay)
- DIISDKExport [RESULT PDMIR\\_GetDelayTTLOut](#) (UINT16 serialNumber, [UINT8](#) \*delay)
- DIISDKExport [RESULT PDMIR\\_GetCounterIntegrationTime](#) (UINT16 serialNumber, [UINT16](#) \*integrationTime)
- DIISDKExport [RESULT PDMIR\\_GetCounterAvalancheStatus](#) (UINT16 serialNumber, [COUNTERSTATUS](#) \*status)
- DIISDKExport [RESULT PDMIR\\_GetCounterValidGateStatus](#) (UINT16 serialNumber, [COUNTERSTATUS](#) \*status)
- DIISDKExport [RESULT PDMIR\\_GetCounterAuxInStatus](#) (UINT16 serialNumber, [COUNTERSTATUS](#) \*status)
- DIISDKExport [RESULT PDMIR\\_GetCounterTriggerInStatus](#) (UINT16 serialNumber, [COUNTERSTATUS](#) \*status)
- DIISDKExport [RESULT PDMIR\\_GetCounterInternalTriggerStatus](#) (UINT16 serialNumber, [COUNTERSTATUS](#) \*status)
- DIISDKExport [RESULT PDMIR\\_GetAllCountersStatus](#) (UINT16 serialNumber, [COUNTERSTATUS](#) \*avalancheStatus, [COUNTERSTATUS](#) \*validGateStatus, [COUNTERSTATUS](#) \*auxInStatus, [COUNTERSTATUS](#) \*triggerInStatus, [COUNTERSTATUS](#) \*internalTriggerStatus)
- DIISDKExport [RESULT PDMIR\\_GetCounterAvalancheLastValue](#) (UINT16 serialNumber, [UINT8](#) \*index, [UINT32](#) \*counts)
- DIISDKExport [RESULT PDMIR\\_GetCounterValidGateLastValue](#) (UINT16 serialNumber, [UINT8](#) \*index, [UINT32](#) \*counts)
- DIISDKExport [RESULT PDMIR\\_GetCounterAuxInLastValue](#) (UINT16 serialNumber, [UINT8](#) \*index, [UINT32](#) \*counts)
- DIISDKExport [RESULT PDMIR\\_GetCounterTriggerInLastValue](#) (UINT16 serialNumber, [UINT8](#) \*index, [UINT32](#) \*counts)
- DIISDKExport [RESULT PDMIR\\_GetCounterInternalTriggerLastValue](#) (UINT16 serialNumber, [UINT8](#) \*index, [UINT32](#) \*counts)
- DIISDKExport [RESULT PDMIR\\_GetAllCountersLastValue](#) (UINT16 serialNumber, [UINT8](#) \*index, [UINT32](#) \*avalancheCounts, [UINT32](#) \*validGateCounts, [UINT32](#) \*auxInCounts, [UINT32](#) \*triggerInCounts, [UINT32](#) \*internalTriggerCounts)
- DIISDKExport [RESULT PDMIR\\_GetCounterAvalancheValueAtIndex](#) (UINT16 serialNumber, [UINT8](#) \*index, [UINT32](#) \*counts)
- DIISDKExport [RESULT PDMIR\\_GetCounterValidGateValueAtIndex](#) (UINT16 serialNumber, [UINT8](#) \*index, [UINT32](#) \*counts)
- DIISDKExport [RESULT PDMIR\\_GetCounterAuxInValueAtIndex](#) (UINT16 serialNumber, [UINT8](#) \*index, [UINT32](#) \*counts)
- DIISDKExport [RESULT PDMIR\\_GetCounterTriggerInValueAtIndex](#) (UINT16 serialNumber, [UINT8](#) \*index, [UINT32](#) \*counts)
- DIISDKExport [RESULT PDMIR\\_GetCounterInternalTriggerValueAtIndex](#) (UINT16 serialNumber, [UINT8](#) \*index, [UINT32](#) \*counts)
- DIISDKExport [RESULT PDMIR\\_GetAllCountersValueAtIndex](#) (UINT16 serialNumber, [UINT8](#) \*index, [UINT32](#) \*avalancheCounts, [UINT32](#) \*validGateCounts, [UINT32](#) \*auxInCounts, [UINT32](#) \*triggerInCounts, [UINT32](#) \*internalTriggerCounts)
- DIISDKExport [RESULT PDMIR\\_GetModuleInfo](#) (UINT16 serialNumber, [MODULEINFO](#) \*info)

- DIISDKExport [RESULT PDMIR\\_GetModuleStatus](#) (UINT16 serialNumber, STATUSBITS \*mStatus, STATUSBITS \*mTemperature, STATUSBITS \*mSpad, STATUSBITS \*mGate, STATUSWARMUP \*mWarm, UINT32 \*errors)
- DIISDKExport void [PDMIR\\_ErrorTranslator](#) (UINT32 error, char \*stringOut)
- DIISDKExport [RESULT PDMIR\\_SaveCurrentConfig](#) (UINT16 serialNumber, UINT8 configNumber, char \*configName)
- DIISDKExport [RESULT PDMIR\\_SaveCurrentConfigForceOverWrite](#) (UINT16 serialNumber, UINT8 configNumber, char \*configName)
- DIISDKExport [RESULT PDMIR\\_DeleteConfig](#) (UINT16 serialNumber, UINT8 configNumber)
- DIISDKExport [RESULT PDMIR\\_SetConfigPowerUp](#) (UINT16 serialNumber, UINT8 configNumber)
- DIISDKExport [RESULT PDMIR\\_LoadConfig](#) (UINT16 serialNumber, UINT8 configNumber)
- DIISDKExport [RESULT PDMIR\\_GetCurrentConfig](#) (UINT16 serialNumber, MODULECONFIG \*config)
- DIISDKExport [RESULT PDMIR\\_GetConfigX](#) (UINT16 serialNumber, UINT8 configNumber, MODULECONFIG \*config, char \*configName)
- DIISDKExport [RESULT PDMIR\\_GetConfigPowerUp](#) (UINT16 serialNumber, UINT8 \*configNumber)

### 7.1.1 Detailed Description

PDM-IR software development kit.

This C header contains all the functions to operate the PDM-IR in user defined applications.