



PDM-IR SDK

PDM-IR

Version 3.1.8

Software Development Kit Manual

April, 2020

Contents

- 1 PDM-IR Software Development Kit (PDM-IR_SDK). 5**

- 2 Module Index 7**
 - 2.1 Modules 7

- 3 Data Structure Index 9**
 - 3.1 Data Structures 9

- 4 File Index 11**
 - 4.1 File List 11

- 5 Module Documentation 13**
 - 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, STATUSBITS *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
6	Data Structure Documentation	53
6.1	MODULECONFIG Struct Reference	53
6.1.1	Detailed Description	53
6.2	MODULEINFO Struct Reference	54
6.2.1	Detailed Description	54
7	File Documentation	55
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:

_EVENT_HANDLE	??
_OVERLAPPED	??
_SECURITY_ATTRIBUTES	??
MODULECONFIG	
Module Configuration Structure	53
MODULEINFO	
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:

PDM-IR_SDK.h	
PDM-IR software development kit	55
WinTypes.h	??

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 runing 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. 

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 MODULEINFO . 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 PDMIR_ErrorTranslator() 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 MODULECONFIG 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 MODULECONFIG 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

- [UUINT16 Amplitude](#)
- [TEMP_VALUES Temperature](#)
- [UUINT32 HoldOff](#)
- [HOLDOFFTYPE HoldOffLevel](#)
- [GATEMODE GateMode](#)
- [GATESHAPE GateShape](#)
- [TREDGE TriggerEdge](#)
- [INT16 TriggerInTh](#)
- [INT16 AuxInTh](#)
- [TRFUNCT TriggerFunction](#)
- [UUINT32 Frequency](#)
- [UUINT32 Ton](#)
- [OUTMODE TtlOut](#)
- [OUTMODE NimOut](#)
- [UUINT8 DelayTriggerIn](#)
- [UUINT8 DelayTriggerInternal](#)
- [UUINT8 DelayAuxIn](#)
- [UUINT8 DelayGate](#)
- [UUINT8 DelayNimOut](#)
- [UUINT8 DelayTtlOut](#)
- [UUINT16 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.