PDMv5 Library

Generated by Doxygen 1.8.16

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

Chapter 2

File Documentation

2.1 pdmv5.h File Reference

Macros

- #define PDMv5_DLL

Functions

- PDMv5_DLL enum sp_return OpenCommunication_PDMv5 (char *portName, struct sp_port **portPointer)

  Open the communication with the device.
- PDMv5_DLL void CloseCommunication_PDMv5 (struct sp_port *port)

  Close the communication with the device.
- PDMv5_DLL int ReadAddress_PDMv5 (struct sp_port *port, unsigned char *address)

  Read the adress of the device.
- PDMv5_DLL int SetAddressSpecific_PDMv5 (struct sp_port *port, unsigned char currentAddress, unsigned char newAddress)

  Set the address of the device.
- PDMv5_DLL int ApplyRequest_PDMv5 (struct sp_port *port, unsigned char address)

  Apply values and parameters (this function must be called in order to really change the state of the PDMv5 card, it can be called after each function or after a list of functions)
- PDMv5_DLL int Save_PDMv5 (struct sp_port *port, unsigned char address)

  Save all values into the device memory.
- PDMv5_DLL int StartCalibration_PDMv5 (struct sp_port *port, unsigned char address, unsigned char calibrationID)

  Start a calibration.
- PDMv5_DLL int GetLIVMeasures_PDMv5 (struct sp_port *port, unsigned char address, unsigned short measureStartIndex, unsigned short measureEndIndex, float *voltages, float *currents, float *powers)

  Get the LIV Measures.
- PDMv5_DLL int GetLIVParameters_PDMv5 (struct sp_port *port, unsigned char address, float *current↩ Minimum, float *currentMaximum, float *currentStep, float *pauseInterval, unsigned char *count, unsigned char *photodiode)

  Get the parameters with which the LIV was done.
- PDMv5_DLL int ReadHardwareType_PDMv5 (struct sp_port *port, unsigned char address, unsigned int *type)

  Read the hardware type.

- PDMv5_DLL int ReadHardwareVersion_PDMv5 (struct sp_port *port, unsigned char address, unsigned char *version)

  Read the hardware version.
- PDMv5_DLL int ReadSerialNumber_PDMv5 (struct sp_port *port, unsigned char address, unsigned int *serial)

  Read the serial number.
- PDMv5_DLL int ReadSoftwareType_PDMv5 (struct sp_port *port, unsigned char address, unsigned int *type)

  Read the software type.
- PDMv5_DLL int ReadSoftwareVersion_PDMv5 (struct sp_port *port, unsigned char address, unsigned char *version)

  Read the software version.
- PDMv5_DLL int ReadSynchro_PDMv5 (struct sp_port *port, unsigned char address, unsigned char *mode)

  Read the synchronisation line which will trigger pulses.
- PDMv5_DLL int ReadDelayLine_PDMv5 (struct sp_port *port, unsigned char address, unsigned char *mode)

  Read the delay line.
- PDMv5_DLL int SetDelayLine_PDMv5 (struct sp_port *port, unsigned char address, unsigned char mode)

  Set the delay line.
- PDMv5_DLL int ReadFrequency_PDMv5 (struct sp_port *port, unsigned char address, unsigned int *frequency)

  Read the frequency (in Hz)
- PDMv5_DLL int SetFrequency_PDMv5 (struct sp_port *port, unsigned char address, unsigned int frequency)

  Set the frequency (in Hz)
- PDMv5_DLL int ReadPulseWidth_PDMv5 (struct sp_port *port, unsigned char address, unsigned int *width)

  Read the pulse width (in ps)
- PDMv5_DLL int SetPulseWidth_PDMv5 (struct sp_port *port, unsigned char address, unsigned int width)

  Set the pulse width (in ps)
- PDMv5_DLL int ReadDelay_PDMv5 (struct sp_port *port, unsigned char address, unsigned int *delay)

  Read the delay (in ps)
- PDMv5_DLL int SetDelay_PDMv5 (struct sp_port *port, unsigned char address, unsigned int delay)

  Set the delay (in ps)
- PDMv5_DLL int ReadCurrent_PDMv5 (struct sp_port *port, unsigned char address, float *current)

  Read the current (in % of maximum current)
- PDMv5_DLL int SetCurrent_PDMv5 (struct sp_port *port, unsigned char address, float current)

  Set the current (in % of maximum current)
- PDMv5_DLL int ReadTemperature_PDMv5 (struct sp_port *port, unsigned char address, float *temperature)

  Read the temperature (in °C)
- PDMv5_DLL int SetTemperature_PDMv5 (struct sp_port *port, unsigned char address, float temperature)

  Set the temperature (in °C)
- PDMv5_DLL int ReadMaximumAverageCurrent_PDMv5 (struct sp_port *port, unsigned char address, float *current)

  Read the mean current limit (in mA)
- PDMv5_DLL int SetMaximumAverageCurrent_PDMv5 (struct sp_port *port, unsigned char address, float current)

  Set the mean current limit (in mA)
- PDMv5_DLL int ReadMaximumCurrent_PDMv5 (struct sp_port *port, unsigned char address, float *current)

  Read the maximum current limit (in mA)
- PDMv5_DLL int SetMaximumCurrent_PDMv5 (struct sp_port *port, unsigned char address, float current)

  Set the maximum current limit (in mA)
- PDMv5_DLL int ReadLaserStatus_PDMv5 (struct sp_port *port, unsigned char address, unsigned char *status)

  Read the laser status.

- PDMv5_DLL int SetLaserStatus_PDMv5 (struct sp_port *port, unsigned char address, unsigned char status)

  Set the laser status.
- PDMv5_DLL int ReadTECMaximumCurrent_PDMv5 (struct sp_port *port, unsigned char address, float *current)

  Read TEC maximum current (in mA)
- PDMv5_DLL int SetTECMaximumCurrent_PDMv5 (struct sp_port *port, unsigned char address, float current)

  Set the TEC maximum current (in mA)
- PDMv5_DLL int ReadTECMaximumVoltage_PDMv5 (struct sp_port *port, unsigned char address, float *voltage)

  Read the TEC maximum voltage (in mV)
- PDMv5_DLL int SetTECMaximumVoltage_PDMv5 (struct sp_port *port, unsigned char address, float voltage)

  Set the TEC maximum voltage (in mV)
- PDMv5_DLL int ReadComplianceVoltage_PDMv5 (struct sp_port *port, unsigned char address, float *voltage)

  Read The Compliance Voltage.
- PDMv5_DLL int SetComplianceVoltage_PDMv5 (struct sp_port *port, unsigned char address, float voltage)

  Set the compliance voltage (in V)
- PDMv5_DLL int ReadOperatingMode_PDMv5 (struct sp_port *port, unsigned char address, unsigned char *mode)

  Read the operating mode.
- PDMv5_DLL int SetOperatingMode_PDMv5 (struct sp_port *port, unsigned char address, unsigned char mode)

  Set the operating mode.
- PDMv5_DLL int ReadAPCPhotodiode_PDMv5 (struct sp_port *port, unsigned char address, unsigned char *mode)

  Read the APC photodiode.
- PDMv5_DLL int SetAPCPhotodiode_PDMv5 (struct sp_port *port, unsigned char address, unsigned char mode)

  Set the APC photodiode.
- PDMv5_DLL int ReadAPCSamplingPeriod_PDMv5 (struct sp_port *port, unsigned char address, unsigned short *time)

  Read the APC sampling period (in ms)
- PDMv5_DLL int SetAPCSamplingPeriod_PDMv5 (struct sp_port *port, unsigned char address, unsigned short time)

  Set the APC sampling period (in ms)
- PDMv5_DLL int ReadAPCHysteresisPercentage_PDMv5 (struct sp_port *port, unsigned char address, float *percentage)

  Read the APC hysteresis percentage (in %)
- PDMv5_DLL int SetAPCHysteresisPercentage_PDMv5 (struct sp_port *port, unsigned char address, float percentage)

  Set the APC hysteresis percentage (in %)
- PDMv5_DLL int ReadPulseLaserStatusPDMv5 (struct sp_port *port, unsigned char address, unsigned char *status)

  Read the pulse laser status.
- PDMv5_DLL int SetPulseLaserStatus_PDMv5 (struct sp_port *port, unsigned char address, unsigned char status)

  Set the pulse laser status.
- PDMv5_DLL int ReadPulseCurrentSource_PDMv5 (struct sp_port *port, unsigned char address, unsigned char *source)

  Read the pulse current source.
- PDMv5_DLL int SetPulseCurrentSource_PDMv5 (struct sp_port *port, unsigned char address, unsigned char source)

Set the pulse current source.

- PDMv5_DLL int ReadPULSE_INDetectionThreshold_PDMv5 (struct sp_port *port, unsigned char address, float *voltage)

  Read the PULSE_IN detection threshold (in mV)
- PDMv5_DLL int SetPULSE_INDetectionThreshold_PDMv5 (struct sp_port *port, unsigned char address, float voltage)

  Set the PULSE_IN detection threshold (in mV)
- PDMv5_DLL int ReadCWLaserStatus_PDMv5 (struct sp_port *port, unsigned char address, unsigned char *status)

  Read the CW laser status.
- PDMv5_DLL int SetCWLaserStatus_PDMv5 (struct sp_port *port, unsigned char address, unsigned char status)

  Set the CW laser status.
- PDMv5_DLL int ReadCWCurrentSource_PDMv5 (struct sp_port *port, unsigned char address, unsigned char *status)

  Read the CW current source.
- PDMv5_DLL int SetCWCurrentSource_PDMv5 (struct sp_port *port, unsigned char address, unsigned char status)

  Set the CW current source.
- PDMv5_DLL int ReadCWMaximumCurrent_PDMv5 (struct sp_port *port, unsigned char address, float *current)

  Read the CW maximum current (in mA)
- PDMv5_DLL int SetCWMaximumCurrent_PDMv5 (struct sp_port *port, unsigned char address, float current)

  Set the CW maximum current (in mA)
- PDMv5_DLL int ReadCWCurrent_PDMv5 (struct sp_port *port, unsigned char address, float *current)

  Read the CW current (in mA)
- PDMv5_DLL int SetCWCurrent_PDMv5 (struct sp_port *port, unsigned char address, float current)

  Set the CW current (in mA)
- PDMv5_DLL int ReadModulationStatus_PDMv5 (struct sp_port *port, unsigned char address, unsigned char *status)

  Read the modulation status.
- PDMv5_DLL int SetModulationStatus_PDMv5 (struct sp_port *port, unsigned char address, unsigned char status)

  Set the modulation status.
- PDMv5_DLL int ReadModulationMaximumCurrent_PDMv5 (struct sp_port *port, unsigned char address, float *current)

  Read the maximum modulation current (in mA)
- PDMv5_DLL int SetModulationMaximumCurrent_PDMv5 (struct sp_port *port, unsigned char address, float current)

  Set the maximum modulation current (in mA)
- PDMv5_DLL int ReadModulationCurrent_PDMv5 (struct sp_port *port, unsigned char address, float *current)

  Read the modulation current (in mA)
- PDMv5_DLL int SetModulationCurrent_PDMv5 (struct sp_port *port, unsigned char address, float current)

  Set the modulation current (in mA)
- PDMv5_DLL int ReadModulationInternalType_PDMv5 (struct sp_port *port, unsigned char address, unsigned char *type)

  Read the internal modulation type.
- PDMv5_DLL int SetModulationInternalType_PDMv5 (struct sp_port *port, unsigned char address, unsigned char type)

  Set the internal modulation type.
- PDMv5_DLL int ReadModulationFrequency_PDMv5 (struct sp_port *port, unsigned char address, float *frequency)

Read the modulation frequency.

- PDMv5_DLL int SetModulationFrequency_PDMv5 (struct sp_port *port, unsigned char address, float frequency)

  Set the modulation frequency (in Hz)
- PDMv5_DLL int ReadModulationExternalCurrentGain_PDMv5 (struct sp_port *port, unsigned char address, float *gain)

  Read the external modulation current gain.
- PDMv5_DLL int SetModulationExternalCurrentGain_PDMv5 (struct sp_port *port, unsigned char address, float gain)

  Set the gain external modulation current.
- PDMv5_DLL int ReadLIVPauseInterval_PDMv5 (struct sp_port *port, unsigned char address, float *time)

  Read the LIV pause interval.
- PDMv5_DLL int SetLIVPauseInterval_PDMv5 (struct sp_port *port, unsigned char address, float time)

  Set the LIV pause interval.
- PDMv5_DLL int ReadLIVMeasuresCount_PDMv5 (struct sp_port *port, unsigned char address, unsigned char *value)

  Read the LIV measures count.
- PDMv5_DLL int SetLIVMeasuresCount_PDMv5 (struct sp_port *port, unsigned char address, unsigned char value)

  Set the LIV measures count.
- PDMv5_DLL int ReadLIVCurrentStep_PDMv5 (struct sp_port *port, unsigned char address, float *current)

  Read the LIV current step (in mA)
- PDMv5_DLL int SetLIVCurrentStep_PDMv5 (struct sp_port *port, unsigned char address, float current)

  Set the LIV current step (in mA)
- PDMv5_DLL int ReadLIVMinimumCurrent_PDMv5 (struct sp_port *port, unsigned char address, float *current)

  Read the LIV minimum current (in mA)
- PDMv5_DLL int SetLIVMinimumCurrent_PDMv5 (struct sp_port *port, unsigned char address, float current)

  Set the LIV minimum current (in mA)
- PDMv5_DLL int ReadLIVMaximumCurrent_PDMv5 (struct sp_port *port, unsigned char address, float *current)

  Read the LIV maximum current (in mA)
- PDMv5_DLL int SetLIVMaximumCurrent_PDMv5 (struct sp_port *port, unsigned char address, float current)

  Set the LIV maximum current (in mA)
- PDMv5_DLL int ReadLIVPhotodiode_PDMv5 (struct sp_port *port, unsigned char address, unsigned char *photodiode)

  Read the LIV photodiode choice.
- PDMv5_DLL int SetLIVPhotodiode_PDMv5 (struct sp_port *port, unsigned char address, unsigned char photodiode)

  Set the LIV photodiode choice.
- PDMv5_DLL int ReadLIVPulseWidth_PDMv5 (struct sp_port *port, unsigned char address, float *time)

  Read the LIV pulse width (in ms)
- PDMv5_DLL int SetLIVPulseWidth_PDMv5 (struct sp_port *port, unsigned char address, float time)

  Set the LIV pulse width (in ms)
- PDMv5_DLL int ReadBFMGain_PDMv5 (struct sp_port *port, unsigned char address, float *gain)

  Read the BFM gain.
- PDMv5_DLL int SetBFMGain_PDMv5 (struct sp_port *port, unsigned char address, float gain)

  Set the BFM gain.
- PDMv5_DLL int ReadPD_EXTGain_PDMv5 (struct sp_port *port, unsigned char address, float *gain)

  Read the PD_EXT photodiode gain.
- PDMv5_DLL int SetPD_EXTGain_PDMv5 (struct sp_port *port, unsigned char address, float gain)

  Set the PD_EXT photodiode gain.

- PDMv5_DLL int ReadExternalMultiElementsBoardCommandVoltage_PDMv5 (struct sp_port *port, unsigned char address, unsigned short *voltage)

  Read the external multielements board command voltage (in mV)
- PDMv5_DLL int SetExternalMultiElementsBoardCommandVoltage_PDMv5 (struct sp_port *port, unsigned char address, unsigned short voltage)

  Set the external multielements board command voltage (in mV)
- PDMv5_DLL int MeasureKeyStatus_PDMv5 (struct sp_port *port, unsigned char address, unsigned char *status)

  Read the key status.
- PDMv5_DLL int MeasureBNCInterlockStatus_PDMv5 (struct sp_port *port, unsigned char address, unsigned char *status)

  Read the BNC interlock status.
- PDMv5_DLL int MeasureEXTInterlockStatus_PDMv5 (struct sp_port *port, unsigned char address, unsigned char *status)

  Read the external interlock status.
- PDMv5_DLL int MeasureLIVStatus_PDMv5 (struct sp_port *port, unsigned char address, unsigned int *percentage)

  Read the LIV status (in %)
- PDMv5_DLL int MeasurePeakCurrentConsign_PDMv5 (struct sp_port *port, unsigned char address, float *current)

  Read the measured peak current consign (in mA)
- PDMv5_DLL int MeasureCWCurrentConsign_PDMv5 (struct sp_port *port, unsigned char address, float *current)

  Read the measured CW consign (in mA)
- PDMv5_DLL int MeasureTemperatureConsign_PDMv5 (struct sp_port *port, unsigned char address, float *temperature)

  Read the measured temperature consign (in °C)
- PDMv5_DLL int MeasureDiodeTemperature_PDMv5 (struct sp_port *port, unsigned char address, float *temperature)

  Read the measured diode temperature (in °C)
- PDMv5_DLL int MeasureTECVoltageReference_PDMv5 (struct sp_port *port, unsigned char address, float *voltage)

  Read the measured TEC Voltage reference (in mV)
- PDMv5_DLL int MeasureTECCurrent_PDMv5 (struct sp_port *port, unsigned char address, float *current)

  Read the measured TEC Current (in mA)
- PDMv5_DLL int MeasureTECExternalTemperature_PDMv5 (struct sp_port *port, unsigned char address, float *temperature)

  Read the measured TEC external temperature (in °C)
- PDMv5_DLL int MeasureMOSTemperature_PDMv5 (struct sp_port *port, unsigned char address, float *temperature)

  Read the measured MOS temperature (in °C)
- PDMv5_DLL int MeasureTECVoltage_PDMv5 (struct sp_port *port, unsigned char address, float *voltage)

  Read the measured TEC Voltage (in mV)
- PDMv5_DLL int MeasureDiodeVoltage_PDMv5 (struct sp_port *port, unsigned char address, float *voltage)

  Read the measured diode voltage (in V)
- PDMv5_DLL int MeasureDiodeCWCurrent_PDMv5 (struct sp_port *port, unsigned char address, float *current)

  Read the measured diode CW current (in mA)
- PDMv5_DLL int MeasureDiodeAverageCurrent_PDMv5 (struct sp_port *port, unsigned char address, float *current)

  Read the measured diode average current (in mA)
- PDMv5_DLL int MeasureDiodePulsedCurrent_PDMv5 (struct sp_port *port, unsigned char address, float *current)

Read the measured diode pulsed current (in mA)

- PDMv5_DLL int MeasureBFMCurrent_PDMv5 (struct sp_port *port, unsigned char address, float *current)

  Read the measured BFM current (in µA)
- PDMv5_DLL int MeasurePD_EXTCurrent_PDMv5 (struct sp_port *port, unsigned char address, float *current)

  Read the measured PD_EXT current (in µA)
- PDMv5_DLL int MeasureCentralDriverEnableStatus_PDMv5 (struct sp_port *port, unsigned char address, unsigned char *status)

  Read the Central Driver enable status.
- PDMv5_DLL int MeasureCentralMMDEnableStatus_PDMv5 (struct sp_port *port, unsigned char address, unsigned char *status)

  Read the Central MMD enable status.
- PDMv5_DLL int MeasurePatchTECExternalTemperature_PDMv5 (struct sp_port *port, unsigned char address, float *temperature)

  Read the measured patch board TEC external temperature (in °C)
- PDMv5_DLL int MeasurePatchPICOLASCurrent_PDMv5 (struct sp_port *port, unsigned char address, float *current)

  Read the measured patch board PICOLAS current (in mA)
- PDMv5_DLL int MeasurePatchPICOLASComplianceVoltage_PDMv5 (struct sp_port *port, unsigned char address, float *voltage)

  Read the measured patch board PICOLAS compliance voltage (in V)
- PDMv5_DLL int MeasurePatchTECCurrent_PDMv5 (struct sp_port *port, unsigned char address, float *current)

  Read the measured patch board TEC current (in mA)
- PDMv5_DLL int MeasurePatchBatteryVoltage_PDMv5 (struct sp_port *port, unsigned char address, float *voltage)

  Read the measured patch board ALIM battery voltage (in V)
- PDMv5_DLL int MeasureInputVoltage_PDMv5 (struct sp_port *port, unsigned char address, float *voltage)

  Read the measured input voltage (in V)
- PDMv5_DLL int MeasureComplianceVoltage_PDMv5 (struct sp_port *port, unsigned char address, float *voltage)

  Read the measured compliance voltage (in V)
- PDMv5_DLL int Measure5VPositiveVoltage_PDMv5 (struct sp_port *port, unsigned char address, float *voltage)

  Read the 5V alimentation voltage (in V)
- PDMv5_DLL int Measure3V3PositiveVoltage_PDMv5 (struct sp_port *port, unsigned char address, float *voltage)

  Read the 3.3V alimentation voltage (in V)
- PDMv5_DLL int Measure2V5PositiveVoltage_PDMv5 (struct sp_port *port, unsigned char address, float *voltage)

  Read the 2.5V alimentation voltage (in V)
- PDMv5_DLL int Measure1V3PositiveVoltage_PDMv5 (struct sp_port *port, unsigned char address, float *voltage)

  Read the 1.3V alimentation voltage (in V)
- PDMv5_DLL int Measure5VNegativeVoltage_PDMv5 (struct sp_port *port, unsigned char address, float *voltage)

  Read the -5V alimentation voltage (in V)
- PDMv5_DLL int MeasureAlarms_PDMv5 (struct sp_port *port, unsigned char address, unsigned int *alarms)

  Read the alarms.

2.1.1 Macro Definition Documentation

2.1.1.1 PDMv5_DLL

```
#define PDMv5_DLL
```

2.1.2 Function Documentation

2.1.2.1 ApplyRequest_PDMv5()

```
PDMv5_DLL int ApplyRequest_PDMv5 (
            struct sp_port * port,
            unsigned char address )
```

Apply values and parameters (this function must be called in order to really change the state of the PDMv5 card, it can be called after each function or after a list of functions)

Parameters

in	*port*	The port used
in	*address*	The address of the device

Returns

An error code or 0

2.1.2.2 CloseCommunication_PDMv5()

```
PDMv5_DLL void CloseCommunication_PDMv5 (
            struct sp_port * port )
```

Close the communication with the device.

Parameters

in	*port*	The port to close

2.1.2.3 GetLIVMeasures_PDMv5()

```
PDMv5_DLL int GetLIVMeasures_PDMv5 (
            struct sp_port * port,
```

```
            unsigned char address,
            unsigned short measureStartIndex,
            unsigned short measureEndIndex,
            float * voltages,
            float * currents,
            float * powers )
```

Get the LIV Measures.

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*measureStartIndex*	The first index of the measures that will be read
in	*measureEndIndex*	The last index of the measures that will be read
out	*voltages*	The voltages measured
out	*currents*	The currents measured
out	*powers*	The powers measured

Returns

An error code or 0

2.1.2.4 GetLIVParameters_PDMv5()

```
PDMv5_DLL int GetLIVParameters_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * currentMinimum,
            float * currentMaximum,
            float * currentStep,
            float * pauseInterval,
            unsigned char * count,
            unsigned char * photodiode )
```

Get the parameters with which the LIV was done.

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*currentMinimum*	The minimum current for the LIV (in mA)
out	*currentMaximum*	The maximum current for the LIV (in mA)
out	*currentStep*	The current step for the LIV (in mA)
out	*pauseInterval*	The pause interval for the LIV (in ms)
out	*count*	The number of the measures
out	*photodiode*	The photodiode used, 0 for BFM, 1 for PD_EXT

Returns

An error code or 0

2.1.2.5 Measure1V3PositiveVoltage_PDMv5()

```
PDMv5_DLL int Measure1V3PositiveVoltage_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * voltage )
```

Read the 1.3V alimentation voltage (in V)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*voltage*	The 1.3V alimentation voltage (in V)

Returns

An error code or 0

2.1.2.6 Measure2V5PositiveVoltage_PDMv5()

```
PDMv5_DLL int Measure2V5PositiveVoltage_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * voltage )
```

Read the 2.5V alimentation voltage (in V)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*voltage*	The 2.5V alimentation voltage (in V)

Returns

An error code or 0

2.1.2.7 Measure3V3PositiveVoltage_PDMv5()

```
PDMv5_DLL int Measure3V3PositiveVoltage_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * voltage )
```

Read the 3.3V alimentation voltage (in V)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*voltage*	The 3.3V alimentation voltage (in V)

Returns

An error code or 0

2.1.2.8 Measure5VNegativeVoltage_PDMv5()

```
PDMv5_DLL int Measure5VNegativeVoltage_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * voltage )
```

Read the -5V alimentation voltage (in V)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*voltage*	The -5V alimentation voltage (in V)

Returns

An error code or 0

2.1.2.9 Measure5VPositiveVoltage_PDMv5()

```
PDMv5_DLL int Measure5VPositiveVoltage_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * voltage )
```

Read the 5V alimentation voltage (in V)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*voltage*	The 5V alimentation voltage (in V)

Returns

An error code or 0

2.1.2.10 MeasureAlarms_PDMv5()

```
PDMv5_DLL int MeasureAlarms_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned int * alarms )
```

Read the alarms.

Each alarm is coded by a bit

- b0 : Interlock
- b1 : Average Current
- b2 : Diode Temperature
- b3 : ICHG Temperature (CW mode) or MOS Temperature (Pulsed mode)
- b4 : TEC Ext
- b5 : BNC Interlock
- b6 : Ext Interlock
- b7 : Key Interlock
- b8 : MOS CW
- b9 : Reserved
- b10 : Open Circuit

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*alarms*	The alarms

Returns

An error code or 0

2.1.2.11 MeasureBFMCurrent_PDMv5()

```
PDMv5_DLL int MeasureBFMCurrent_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * current )
```

Read the measured BFM current (in µA)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*current*	The measured BFM current (in µA)

Returns

An error code or 0

2.1.2.12 MeasureBNCInterlockStatus_PDMv5()

```
PDMv5_DLL int MeasureBNCInterlockStatus_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned char * status )
```

Read the BNC interlock status.

0 : OFF, 1 : ON

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*status*	The BNC interlock status

Returns

An error code or 0

2.1.2.13 MeasureCentralDriverEnableStatus_PDMv5()

```
PDMv5_DLL int MeasureCentralDriverEnableStatus_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned char * status )
```

Read the Central Driver enable status.

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*status*	The Central Driver enable status

Returns

An error code or 0

2.1.2.14 MeasureCentralMMDEnableStatus_PDMv5()

```
PDMv5_DLL int MeasureCentralMMDEnableStatus_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned char * status )
```

Read the Central MMD enable status.

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*status*	The Central MMD enable status

Returns

An error code or 0

2.1.2.15 MeasureComplianceVoltage_PDMv5()

```
PDMv5_DLL int MeasureComplianceVoltage_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * voltage )
```

Read the measured compliance voltage (in V)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*voltage*	The measured compliance voltage (in V)

Returns

An error code or 0

2.1.2.16 MeasureCWCurrentConsign_PDMv5()

```
PDMv5_DLL int MeasureCWCurrentConsign_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * current )
```

Read the measured CW consign (in mA)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*current*	The measured CW consign (in mA)

Returns

An error code or 0

2.1.2.17 MeasureDiodeAverageCurrent_PDMv5()

```
PDMv5_DLL int MeasureDiodeAverageCurrent_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * current )
```

Read the measured diode average current (in mA)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*current*	The measured diode average current (in mA)

Returns

An error code or 0

2.1.2.18 MeasureDiodeCWCurrent_PDMv5()

```
PDMv5_DLL int MeasureDiodeCWCurrent_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * current )
```

Read the measured diode CW current (in mA)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*current*	The measured diode CW current (in mA)

Returns

An error code or 0

2.1.2.19 MeasureDiodePulsedCurrent_PDMv5()

```
PDMv5_DLL int MeasureDiodePulsedCurrent_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * current )
```

Read the measured diode pulsed current (in mA)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*current*	The measured diode pulsed current (in mA)

Returns

An error code or 0

2.1.2.20 MeasureDiodeTemperature_PDMv5()

```
PDMv5_DLL int MeasureDiodeTemperature_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * temperature )
```

Read the measured diode temperature (in °C)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*temperature*	The measured diode temperature (in °C)

Returns

An error code or 0

2.1.2.21 MeasureDiodeVoltage_PDMv5()

```
PDMv5_DLL int MeasureDiodeVoltage_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * voltage )
```

Read the measured diode voltage (in V)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*voltage*	The measured diode voltage (in V)

Returns

An error code or 0

2.1.2.22 MeasureEXTInterlockStatus_PDMv5()

```
PDMv5_DLL int MeasureEXTInterlockStatus_PDMv5 (
            struct sp_port * port,
```

```
            unsigned char address,
            unsigned char * status )
```

Read the external interlock status.

0 : OFF, 1 : ON

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*status*	The external interlock status

Returns

An error code or 0

2.1.2.23 MeasureInputVoltage_PDMv5()

```
PDMv5_DLL int MeasureInputVoltage_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * voltage )
```

Read the measured input voltage (in V)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*voltage*	The measured input voltage (in V)

Returns

An error code or 0

2.1.2.24 MeasureKeyStatus_PDMv5()

```
PDMv5_DLL int MeasureKeyStatus_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned char * status )
```

Read the key status.

0 : UNTURNED, 1 : TURNED

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*status*	The key status

Returns

An error code or 0

2.1.2.25 MeasureLIVStatus_PDMv5()

```
PDMv5_DLL int MeasureLIVStatus_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned int * percentage )
```

Read the LIV status (in %)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*percentage*	The LIV status (in %)

Returns

An error code or 0

2.1.2.26 MeasureMOSTemperature_PDMv5()

```
PDMv5_DLL int MeasureMOSTemperature_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * temperature )
```

Read the measured MOS temperature (in °C)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*temperature*	The measured MOS temperature (in °C)

Returns

An error code or 0

2.1.2.27 MeasurePatchBatteryVoltage_PDMv5()

```
PDMv5_DLL int MeasurePatchBatteryVoltage_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * voltage )
```

Read the measured patch board ALIM battery voltage (in V)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*voltage*	The measured patch board ALIM battery voltage (in V)

Returns

An error code or 0

2.1.2.28 MeasurePatchPICOLASComplianceVoltage_PDMv5()

```
PDMv5_DLL int MeasurePatchPICOLASComplianceVoltage_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * voltage )
```

Read the measured patch board PICOLAS compliance voltage (in V)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*voltage*	The measured patch board PICOLAS compliance voltage (in V)

Returns

An error code or 0

2.1.2.29 MeasurePatchPICOLASCurrent_PDMv5()

```
PDMv5_DLL int MeasurePatchPICOLASCurrent_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * current )
```

Read the measured patch board PICOLAS current (in mA)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*current*	The measured patch board PICOLAS current (in mA)

Returns

An error code or 0

2.1.2.30 MeasurePatchTECCurrent_PDMv5()

```
PDMv5_DLL int MeasurePatchTECCurrent_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * current )
```

Read the measured patch board TEC current (in mA)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*current*	The measured patch board TEC current (in mA)

Returns

An error code or 0

2.1.2.31 MeasurePatchTECExternalTemperature_PDMv5()

```
PDMv5_DLL int MeasurePatchTECExternalTemperature_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * temperature )
```

Read the measured patch board TEC external temperature (in °C)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*temperature*	The measured patch board TEC external temperature (in °C)

Returns

An error code or 0

2.1.2.32 MeasurePD_EXTCurrent_PDMv5()

```
PDMv5_DLL int MeasurePD_EXTCurrent_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * current )
```

Read the measured PD_EXT current (in µA)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*current*	The measured PD_EXT current (in µA)

Returns

An error code or 0

2.1.2.33 MeasurePeakCurrentConsign_PDMv5()

```
PDMv5_DLL int MeasurePeakCurrentConsign_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * current )
```

Read the measured peak current consign (in mA)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*current*	The measured peak current consign (in mA)

Returns

An error code or 0

2.1.2.34 MeasureTECCurrent_PDMv5()

```
PDMv5_DLL int MeasureTECCurrent_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * current )
```

Read the measured TEC Current (in mA)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*current*	The measured TEC Current (in mA)

Returns

An error code or 0

2.1.2.35 MeasureTECExternalTemperature_PDMv5()

```
PDMv5_DLL int MeasureTECExternalTemperature_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * temperature )
```

Read the measured TEC external temperature (in °C)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*temperature*	The measured TEC external temperature (in °C)

Returns

An error code or 0

2.1.2.36 MeasureTECVoltage_PDMv5()

```
PDMv5_DLL int MeasureTECVoltage_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * voltage )
```

Read the measured TEC Voltage (in mV)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*voltage*	The measured TEC Voltage (in mV)

Returns

An error code or 0

2.1.2.37 MeasureTECVoltageReference_PDMv5()

```
PDMv5_DLL int MeasureTECVoltageReference_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * voltage )
```

Read the measured TEC Voltage reference (in mV)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*voltage*	The measured TEC Voltage reference (in mV)

Returns

An error code or 0

2.1.2.38 MeasureTemperatureConsign_PDMv5()

```
PDMv5_DLL int MeasureTemperatureConsign_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * temperature )
```

Read the measured temperature consign (in °C)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*temperature*	The measured temperature consign (in °C)

Returns

An error code or 0

2.1.2.39 OpenCommunication_PDMv5()

```
PDMv5_DLL enum sp_return OpenCommunication_PDMv5 (
            char * portName,
            struct sp_port ** portPointer )
```

Open the communication with the device.

Parameters

in	*portName*	The name of the COM port
out	*portPointer*	A pointer to a port that will be initialized correctly

Returns

A value indicating the status error (RS232 error code) = 0 if no error

2.1.2.40 ReadAddress_PDMv5()

```
PDMv5_DLL int ReadAddress_PDMv5 (
            struct sp_port * port,
            unsigned char * address )
```

Read the adress of the device.

Parameters

in	*port*	The port used
out	*address*	The address of the device

Returns

An error code or 0

2.1.2.41 ReadAPCHysteresisPercentage_PDMv5()

```
PDMv5_DLL int ReadAPCHysteresisPercentage_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * percentage )
```

Read the APC hysteresis percentage (in %)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*percentage*	The APC hysteresis percentage (in %)

Returns

An error code or 0

2.1.2.42 ReadAPCPhotodiode_PDMv5()

```
PDMv5_DLL int ReadAPCPhotodiode_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned char * mode )
```

Read the APC photodiode.

0 : BFM, 1 : PD_EXT

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*mode*	The APC photodiode

Returns

An error code or 0

2.1.2.43 ReadAPCSamplingPeriod_PDMv5()

```
PDMv5_DLL int ReadAPCSamplingPeriod_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned short * time )
```

Read the APC sampling period (in ms)

0 : BFM, 1 : PD_EXT

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*time*	The APC sampling period (in ms)

Returns

An error code or 0

2.1.2.44 ReadBFMGain_PDMv5()

```
PDMv5_DLL int ReadBFMGain_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * gain )
```

Read the BFM gain.

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*gain*	The BFM gain

Returns

An error code or 0

2.1.2.45 ReadComplianceVoltage_PDMv5()

```
PDMv5_DLL int ReadComplianceVoltage_PDMv5 (
            struct sp_port * port,
```

```
            unsigned char address,
            float * voltage )
```

Read The Compliance Voltage.

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*voltage*	The Compliance Voltage

Returns

An error code or 0

2.1.2.46 ReadCurrent_PDMv5()

```
PDMv5_DLL int ReadCurrent_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * current )
```

Read the current (in % of maximum current)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*current*	The current (in % of maximum current)

Returns

An error code or 0

2.1.2.47 ReadCWCurrent_PDMv5()

```
PDMv5_DLL int ReadCWCurrent_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * current )
```

Read the CW current (in mA)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*current*	The CW current (in mA)

Returns

An error code or 0

2.1.2.48 ReadCWCurrentSource_PDMv5()

```
PDMv5_DLL int ReadCWCurrentSource_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned char * status )
```

Read the CW current source.

0 : Internal DAC; 1 : Potentiometer POT; 2 : External (SMA or connector)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*status*	The CW current source

Returns

An error code or 0

2.1.2.49 ReadCWLaserStatus_PDMv5()

```
PDMv5_DLL int ReadCWLaserStatus_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned char * status )
```

Read the CW laser status.

0 : OFF, 1 : ON

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*status*	The CW laser status

Returns

An error code or 0

2.1.2.50 ReadCWMaximumCurrent_PDMv5()

```
PDMv5_DLL int ReadCWMaximumCurrent_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * current )
```

Read the CW maximum current (in mA)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*current*	The CW maximum current (in mA)

Returns

An error code or 0

2.1.2.51 ReadDelay_PDMv5()

```
PDMv5_DLL int ReadDelay_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned int * delay )
```

Read the delay (in ps)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*delay*	The delay (in ps)

Returns

An error code or 0

2.1.2.52 ReadDelayLine_PDMv5()

```
PDMv5_DLL int ReadDelayLine_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned char * mode )
```

Read the delay line.

0 : NONE (SMA TTL/LVTTL input, duration of pulses will be the same as the applied signal), 1: Internal

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*mode*	The delay line

Returns

An error code or 0

2.1.2.53 ReadExternalMultiElementsBoardCommandVoltage_PDMv5()

```
PDMv5_DLL int ReadExternalMultiElementsBoardCommandVoltage_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned short * voltage )
```

Read the external multielements board command voltage (in mV)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*voltage*	The external multielements board command voltage (in mV)

Returns

An error code or 0

2.1.2.54 ReadFrequency_PDMv5()

```
PDMv5_DLL int ReadFrequency_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned int * frequency )
```

Read the frequency (in Hz)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*frequency*	The frequency (in Hz)

Returns

An error code or 0

2.1.2.55 ReadHardwareType_PDMv5()

```
PDMv5_DLL int ReadHardwareType_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned int * type )
```

Read the hardware type.

- 'A' is for PDM
- 'B' is for MMD
- 'C' is for Cristal
- 'D' is for Central
- 'E' is for Pulsepicker
- 'F' is for Shaper
- 'G' is for PDMv5

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*type*	An integer corresponding to the hardware type

Returns

An error code or 0

2.1.2.56 ReadHardwareVersion_PDMv5()

```
PDMv5_DLL int ReadHardwareVersion_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned char * version )
```

Read the hardware version.

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*version*	Three integers corresponding to the hardware version

Returns

An error code or 0

2.1.2.57 ReadLaserStatus_PDMv5()

```
PDMv5_DLL int ReadLaserStatus_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned char * status )
```

Read the laser status.

0 for OFF, 1 for ON

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*status*	The laser status

Returns

An error code or 0

2.1.2.58 ReadLIVCurrentStep_PDMv5()

```
PDMv5_DLL int ReadLIVCurrentStep_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * current )
```

Read the LIV current step (in mA)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*current*	The LIV current step (in mA)

Returns

An error code or 0

2.1.2.59 ReadLIVMaximumCurrent_PDMv5()

```
PDMv5_DLL int ReadLIVMaximumCurrent_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * current )
```

Read the LIV maximum current (in mA)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*current*	The LIV maximum current (in mA)

Returns

An error code or 0

2.1.2.60 ReadLIVMeasuresCount_PDMv5()

```
PDMv5_DLL int ReadLIVMeasuresCount_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned char * value )
```

Read the LIV measures count.

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*value*	The LIV measures count

Returns

An error code or 0

2.1.2.61 ReadLIVMinimumCurrent_PDMv5()

```
PDMv5_DLL int ReadLIVMinimumCurrent_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * current )
```

Read the LIV minimum current (in mA)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*current*	The LIV minimum current (in mA)

Returns

An error code or 0

2.1.2.62 ReadLIVPauseInterval_PDMv5()

```
PDMv5_DLL int ReadLIVPauseInterval_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * time )
```

Read the LIV pause interval.

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*time*	the LIV pause interval (in ms)

Returns

An error code or 0

2.1.2.63 ReadLIVPhotodiode_PDMv5()

```
PDMv5_DLL int ReadLIVPhotodiode_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned char * photodiode )
```

Read the LIV photodiode choice.

0 for BFM, 1 for PD_EXT

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*photodiode*	The LIV photodiode choice

Returns

An error code or 0

2.1.2.64 ReadLIVPulseWidth_PDMv5()

```
PDMv5_DLL int ReadLIVPulseWidth_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * time )
```

Read the LIV pulse width (in ms)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*time*	The LIV pulse width (in ms)

Returns

An error code or 0

2.1.2.65 ReadMaximumAverageCurrent_PDMv5()

```
PDMv5_DLL int ReadMaximumAverageCurrent_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * current )
```

Read the mean current limit (in mA)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*current*	The mean current limit (in mA)

Returns

An error code or 0

2.1.2.66 ReadMaximumCurrent_PDMv5()

```
PDMv5_DLL int ReadMaximumCurrent_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * current )
```

Read the maximum current limit (in mA)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*current*	The maximum current limit (in mA)

Returns

An error code or 0

2.1.2.67 ReadModulationCurrent_PDMv5()

```
PDMv5_DLL int ReadModulationCurrent_PDMv5 (
            struct sp_port * port,
```

```
            unsigned char address,
            float * current )
```

Read the modulation current (in mA)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*current*	The modulation current (in mA)

Returns

An error code or 0

2.1.2.68 ReadModulationExternalCurrentGain_PDMv5()

```
PDMv5_DLL int ReadModulationExternalCurrentGain_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * gain )
```

Read the external modulation current gain.

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*gain*	The external modulation current gain (in mA/V)

Returns

An error code or 0

2.1.2.69 ReadModulationFrequency_PDMv5()

```
PDMv5_DLL int ReadModulationFrequency_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * frequency )
```

Read the modulation frequency.

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*frequency*	The modulation frequency (in Hz)

Returns

An error code or 0

2.1.2.70 ReadModulationInternalType_PDMv5()

```
PDMv5_DLL int ReadModulationInternalType_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned char * type )
```

Read the internal modulation type.

0 : Sinusoidal, 1 : Triangular, 2 : Square

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*type*	The internal modulation type

Returns

An error code or 0

2.1.2.71 ReadModulationMaximumCurrent_PDMv5()

```
PDMv5_DLL int ReadModulationMaximumCurrent_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * current )
```

Read the maximum modulation current (in mA)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*current*	The maximum modulation current (in mA)

Returns

An error code or 0

2.1.2.72 ReadModulationStatus_PDMv5()

```
PDMv5_DLL int ReadModulationStatus_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned char * status )
```

Read the modulation status.

0 : Off; 1 : On

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*status*	modulation status

Returns

An error code or 0

2.1.2.73 ReadOperatingMode_PDMv5()

```
PDMv5_DLL int ReadOperatingMode_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned char * mode )
```

Read the operating mode.

0 : ACC, 1 : APC

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*mode*	The operating mode

Returns

An error code or 0

2.1.2.74 ReadPD_EXTGain_PDMv5()

```
PDMv5_DLL int ReadPD_EXTGain_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * gain )
```

Read the PD_EXT photodiode gain.

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*gain*	The PD_EXT photodiode gain

Returns

An error code or 0

2.1.2.75 ReadPULSE_INDetectionThreshold_PDMv5()

```
PDMv5_DLL int ReadPULSE_INDetectionThreshold_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * voltage )
```

Read the PULSE_IN detection threshold (in mV)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*voltage*	The PULSE_IN detection threshold (in mV)

Returns

An error code or 0

2.1.2.76 ReadPulseCurrentSource_PDMv5()

```
PDMv5_DLL int ReadPulseCurrentSource_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned char * source )
```

Read the pulse current source.

0 : Internal DAC; 1 : Potentiometer POT; 2 : External (SMA or connector)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*source*	The pulse current source

Returns

An error code or 0

2.1.2.77 ReadPulseLaserStatusPDMv5()

```
PDMv5_DLL int ReadPulseLaserStatusPDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned char * status )
```

Read the pulse laser status.

0 : OFF, 1 : ON

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*status*	The pulsedd mode laser status

Returns

An error code or 0

2.1.2.78 ReadPulseWidth_PDMv5()

```
PDMv5_DLL int ReadPulseWidth_PDMv5 (
            struct sp_port * port,
```

```
            unsigned char address,
            unsigned int * width )
```

Read the pulse width (in ps)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*width*	The pulse width (in ps)

Returns

An error code or 0

2.1.2.79 ReadSerialNumber_PDMv5()

```
PDMv5_DLL int ReadSerialNumber_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned int * serial )
```

Read the serial number.

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*serial*	An integer corresponding to the serial number

Returns

An error code or 0

2.1.2.80 ReadSoftwareType_PDMv5()

```
PDMv5_DLL int ReadSoftwareType_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned int * type )
```

Read the software type.

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*type*	An integer corresponding to the software type

Returns

An error code or 0

2.1.2.81 ReadSoftwareVersion_PDMv5()

```
PDMv5_DLL int ReadSoftwareVersion_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned char * version )
```

Read the software version.

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*version*	3 integers corresponding to the software version

Returns

An error code or 0

2.1.2.82 ReadSynchro_PDMv5()

```
PDMv5_DLL int ReadSynchro_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned char * mode )
```

Read the synchronisation line which will trigger pulses.

0 : External TTL/LVTTL (SMA input), 1: External LVDS (optional SMA inputs), 2 : Internal clock

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*mode*	The synchronisation line

Returns

An error code or 0

2.1.2.83 ReadTECMaximumCurrent_PDMv5()

```
PDMv5_DLL int ReadTECMaximumCurrent_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * current )
```

Read TEC maximum current (in mA)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*current*	The TEC maximum current (in mA)

Returns

An error code or 0

2.1.2.84 ReadTECMaximumVoltage_PDMv5()

```
PDMv5_DLL int ReadTECMaximumVoltage_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * voltage )
```

Read the TEC maximum voltage (in mV)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*voltage*	The TEC maximum voltage (in mV)

Returns

An error code or 0

2.1.2.85 ReadTemperature_PDMv5()

```
PDMv5_DLL int ReadTemperature_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float * temperature )
```

Read the temperature (in ℃)

Parameters

in	*port*	The port used
in	*address*	The address of the device
out	*temperature*	The temperature (in ℃)

Returns

An error code or 0

2.1.2.86 Save_PDMv5()

```
PDMv5_DLL int Save_PDMv5 (
            struct sp_port * port,
            unsigned char address )
```

Save all values into the device memory.

Parameters

in	*port*	The port used
in	*address*	The address of the device

Returns

An error code or 0

2.1.2.87 SetAddressSpecific_PDMv5()

```
PDMv5_DLL int SetAddressSpecific_PDMv5 (
            struct sp_port * port,
            unsigned char currentAddress,
            unsigned char newAddress )
```

Set the address of the device.

Parameters

in	*port*	The port used
in	*currentAddress*	The current address of the device
in	*newAddress*	The new address of the device

Returns

An error code or 0

2.1.2.88 SetAPCHysteresisPercentage_PDMv5()

```
PDMv5_DLL int SetAPCHysteresisPercentage_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float percentage )
```

Set the APC hysteresis percentage (in %)

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*percentage*	The APC hysteresis percentage (in %)

Returns

An error code or 0

2.1.2.89 SetAPCPhotodiode_PDMv5()

```
PDMv5_DLL int SetAPCPhotodiode_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned char mode )
```

Set the APC photodiode.

0 : BFM, 1 : PD_EXT

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*mode*	The APC photodiode

Returns

An error code or 0

2.1.2.90 SetAPCSamplingPeriod_PDMv5()

```
PDMv5_DLL int SetAPCSamplingPeriod_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned short time )
```

Set the APC sampling period (in ms)

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*time*	The APC sampling period (in ms)

Returns

An error code or 0

2.1.2.91 SetBFMGain_PDMv5()

```
PDMv5_DLL int SetBFMGain_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float gain )
```

Set the BFM gain.

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*gain*	The BFM gain

Returns

An error code or 0

2.1.2.92 SetComplianceVoltage_PDMv5()

```
PDMv5_DLL int SetComplianceVoltage_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float voltage )
```

Set the compliance voltage (in V)

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*voltage*	The compliance voltage (in V)

Returns

An error code or 0

2.1.2.93 SetCurrent_PDMv5()

```
PDMv5_DLL int SetCurrent_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float current )
```

Set the current (in % of maximum current)

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*current*	The current (in % of maximum current)

Returns

An error code or 0

2.1.2.94 SetCWCurrent_PDMv5()

```
PDMv5_DLL int SetCWCurrent_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float current )
```

Set the CW current (in mA)

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*current*	The CW current (in mA)

Returns

An error code or 0

2.1.2.95 SetCWCurrentSource_PDMv5()

```
PDMv5_DLL int SetCWCurrentSource_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned char status )
```

Set the CW current source.

0 : Internal DAC; 1 : Potentiometer POT; 2 : External (SMA or connector)

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*status*	The CW current source

Returns

An error code or 0

2.1.2.96 SetCWLaserStatus_PDMv5()

```
PDMv5_DLL int SetCWLaserStatus_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned char status )
```

Set the CW laser status.

0 : OFF, 1 : ON

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*status*	The CW laser status

Returns

An error code or 0

2.1.2.97 SetCWMaximumCurrent_PDMv5()

```
PDMv5_DLL int SetCWMaximumCurrent_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float current )
```

Set the CW maximum current (in mA)

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*current*	The CW maximum current (in mA)

Returns

An error code or 0

2.1.2.98 SetDelay_PDMv5()

```
PDMv5_DLL int SetDelay_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned int delay )
```

Set the delay (in ps)

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*delay*	The delay (in ps)

Returns

An error code or 0

2.1.2.99 SetDelayLine_PDMv5()

```
PDMv5_DLL int SetDelayLine_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned char mode )
```

Set the delay line.

0 : NONE (SMA TTL/LVTTL input, duration of pulses will be the same as the applied signal), 1: Internal

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*mode*	The delay line

Returns

An error code or 0

2.1.2.100 SetExternalMultiElementsBoardCommandVoltage_PDMv5()

```
PDMv5_DLL int SetExternalMultiElementsBoardCommandVoltage_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned short voltage )
```

Set the external multielements board command voltage (in mV)

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*voltage*	The external multielements board command voltage (in mV)

Returns

An error code or 0

2.1.2.101 SetFrequency_PDMv5()

```
PDMv5_DLL int SetFrequency_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned int frequency )
```

Set the frequency (in Hz)

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*frequency*	The frequency (in Hz)

Returns

An error code or 0

2.1.2.102 SetLaserStatus_PDMv5()

```
PDMv5_DLL int SetLaserStatus_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned char status )
```

Set the laser status.

0 for OFF, 1 for ON

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*status*	The laser status

Returns

An error code or 0

2.1.2.103 SetLIVCurrentStep_PDMv5()

```
PDMv5_DLL int SetLIVCurrentStep_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float current )
```

Set the LIV current step (in mA)

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*current*	The LIV current step (in mA)

Returns

An error code or 0

2.1.2.104 SetLIVMaximumCurrent_PDMv5()

```
PDMv5_DLL int SetLIVMaximumCurrent_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float current )
```

Set the LIV maximum current (in mA)

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*current*	The LIV maximum current (in mA)

Returns

An error code or 0

2.1.2.105 SetLIVMeasuresCount_PDMv5()

```
PDMv5_DLL int SetLIVMeasuresCount_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned char value )
```

Set the LIV measures count.

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*value*	The LIV measures count

Returns

An error code or 0

2.1.2.106 SetLIVMinimumCurrent_PDMv5()

```
PDMv5_DLL int SetLIVMinimumCurrent_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float current )
```

Set the LIV minimum current (in mA)

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*current*	The LIV minimum current (in mA)

Returns

An error code or 0

2.1.2.107 SetLIVPauseInterval_PDMv5()

```
PDMv5_DLL int SetLIVPauseInterval_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float time )
```

Set the LIV pause interval.

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*time*	The LIV pause interval (in ms)

Returns

An error code or 0

2.1.2.108 SetLIVPhotodiode_PDMv5()

```
PDMv5_DLL int SetLIVPhotodiode_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned char photodiode )
```

Set the LIV photodiode choice.

0 for BFM, 1 for PD_EXT

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*photodiode*	The LIV photodiode choice

Returns

An error code or 0

2.1.2.109 SetLIVPulseWidth_PDMv5()

```
PDMv5_DLL int SetLIVPulseWidth_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float time )
```

Set the LIV pulse width (in ms)

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*time*	The LIV pulse width (in ms)

Returns

An error code or 0

2.1.2.110 SetMaximumAverageCurrent_PDMv5()

```
PDMv5_DLL int SetMaximumAverageCurrent_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float current )
```

Set the mean current limit (in mA)

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*current*	The mean current limit (in mA)

Returns

An error code or 0

2.1.2.111 SetMaximumCurrent_PDMv5()

```
PDMv5_DLL int SetMaximumCurrent_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float current )
```

Set the maximum current limit (in mA)

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*current*	The maximum current limit (in mA)

Returns

An error code or 0

2.1.2.112 SetModulationCurrent_PDMv5()

```
PDMv5_DLL int SetModulationCurrent_PDMv5 (
            struct sp_port * port,
```

```
            unsigned char address,
            float current )
```

Set the modulation current (in mA)

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*current*	The modulation current (in mA)

Returns

An error code or 0

2.1.2.113 SetModulationExternalCurrentGain_PDMv5()

```
PDMv5_DLL int SetModulationExternalCurrentGain_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float gain )
```

Set the gain external modulation current.

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*gain*	The gain external modulation current (in mA/V)

Returns

An error code or 0

2.1.2.114 SetModulationFrequency_PDMv5()

```
PDMv5_DLL int SetModulationFrequency_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float frequency )
```

Set the modulation frequency (in Hz)

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*frequency*	The modulation frequency (in Hz)

Returns

An error code or 0

2.1.2.115 SetModulationInternalType_PDMv5()

```
PDMv5_DLL int SetModulationInternalType_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned char type )
```

Set the internal modulation type.

0 : Sinusoidal, 1 : Triangular, 2 : Square

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*type*	The internal modulation type

Returns

An error code or 0

2.1.2.116 SetModulationMaximumCurrent_PDMv5()

```
PDMv5_DLL int SetModulationMaximumCurrent_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float current )
```

Set the maximum modulation current (in mA)

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*current*	The maximum modulation current (in mA)

Returns

An error code or 0

2.1.2.117 SetModulationStatus_PDMv5()

```
PDMv5_DLL int SetModulationStatus_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned char status )
```

Set the modulation status.

0 : Off; 1 : On

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*status*	The modulation status

Returns

An error code or 0

2.1.2.118 SetOperatingMode_PDMv5()

```
PDMv5_DLL int SetOperatingMode_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned char mode )
```

Set the operating mode.

0 : ACC, 1 : APC

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*mode*	The operating mode

Returns

An error code or 0

2.1.2.119 SetPD_EXTGain_PDMv5()

```
PDMv5_DLL int SetPD_EXTGain_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float gain )
```

Set the PD_EXT photodiode gain.

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*gain*	The PD_EXT photodiode gain

Returns

An error code or 0

2.1.2.120 SetPULSE_INDetectionThreshold_PDMv5()

```
PDMv5_DLL int SetPULSE_INDetectionThreshold_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float voltage )
```

Set the PULSE_IN detection threshold (in mV)

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*voltage*	The PULSE_IN detection threshold (in mV)

Returns

An error code or 0

2.1.2.121 SetPulseCurrentSource_PDMv5()

```
PDMv5_DLL int SetPulseCurrentSource_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned char source )
```

Set the pulse current source.

0 : Internal DAC; 1 : Potentiometer POT; 2 : External (SMA or connector)

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*source*	The pulse current source

Returns

An error code or 0

2.1.2.122 SetPulseLaserStatus_PDMv5()

```
PDMv5_DLL int SetPulseLaserStatus_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned char status )
```

Set the pulse laser status.

0 : OFF, 1 : ON

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*status*	The pulse laser status

Returns

An error code or 0

2.1.2.123 SetPulseWidth_PDMv5()

```
PDMv5_DLL int SetPulseWidth_PDMv5 (
            struct sp_port * port,
```

```
            unsigned char address,
            unsigned int width )
```

Set the pulse width (in ps)

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*width*	The pulse width (in ps)

Returns

An error code or 0

2.1.2.124 SetTECMaximumCurrent_PDMv5()

```
PDMv5_DLL int SetTECMaximumCurrent_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float current )
```

Set the TEC maximum current (in mA)

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*current*	The TEC maximum current (in mA)

Returns

An error code or 0

2.1.2.125 SetTECMaximumVoltage_PDMv5()

```
PDMv5_DLL int SetTECMaximumVoltage_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float voltage )
```

Set the TEC maximum voltage (in mV)

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*voltage*	The TEC maximum voltage (in mV)

Returns

An error code or 0

2.1.2.126 SetTemperature_PDMv5()

```
PDMv5_DLL int SetTemperature_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            float temperature )
```

Set the temperature (in °C)

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*temperature*	The temperature (in °C)

Returns

An error code or 0

2.1.2.127 StartCalibration_PDMv5()

```
PDMv5_DLL int StartCalibration_PDMv5 (
            struct sp_port * port,
            unsigned char address,
            unsigned char calibrationID )
```

Start a calibration.

Once a calibration is started, all commands will return an error code -30032 until the calibration is finished Calibration IDs are :

- 0x00 : RESERVED
- 0x01 : Measure LIV (Light, Intensity, Voltage)
- 0x02 : Measure IP (Intensity, Power)
- 0x03 : Measure IV (Intensity, Voltage)

Parameters

in	*port*	The port used
in	*address*	The address of the device
in	*calibrationID*	The identifier of the calibration

Returns

An error code or 0

Index