Aemulus

AM400e .NET Wrapper Library

Programming Manual

Version 1.5, 02-2014

Table of Contents

Section 1: Pre-requisites

Test program needs to add following references:

Aemulus.Hardware.SMU.dll

Section 2: General

SMU

Synopsis

SMU (HardwareProfile, testHead, testSite, offline)

Arguments

String ^ HardwareProfile (in)

Specifies the instrument handle.

int testHead (in)

Specifies the test head number.

int testSite (in)

Specifies the test site number.

bool offline (in)

Specifies whether to run tester in offline mode.

Descriptions

This function will return a *SMU* object to the test program upon initialization of resource mapping.

Example C++/CLI codes:

```
SMU ^ smu = nullptr;
if (smu == nullptr)
{
	try
	{
		smu = gcnew SMU("C:\\LX5511.amap", 0, 0, false);
	}
	catch (Exception ^ex)
	{
		return 0;
	}
}
```

Once *SMU* object is returned, users can then access all hardware functions. Examples:

```
smu->SetBandwidth("Vcc", 0);
smu->ClampCurrent("Vcc", 0.5);
smu->DriveVoltage("Vcc", 3.3);
smu->OnSmuPin("Vcc", true, false);
```

When this object is disposed, the communication session will be closed accordingly.

Core Library Function Mapping

1. AemDCPwr_InitChannels
2. AemDCPwr_Close

SMU

Synopsis

SMU (HardwareProfile, testHead, testSite, offline, initOption)

Arguments

String ^ HardwareProfile (in)

Specifies the instrument handle.

int testHead (in)

Specifies the test head number.

int testSite (in)

Specifies the test site number.

bool offline (in)

Specifies whether to run tester in offline mode.

int initOption (in)

SMU initialization option bits.

Descriptions

This function will return a *SMU* object to the test program upon initialization of resource mapping.

Example C++/CLI codes:

```
SMU ^ smu = nullptr;
if (smu == nullptr)
{
	try
	{
		smu = gcnew SMU("C:\\LX5511.amap", 0, 0, false, 0xF);
	}
	catch (Exception ^ex)
	{
		return 0;
```

```
        }
}
```

Once *SMU* object is returned, users can then access all hardware functions. Examples:

```
smu->SetBandwidth("Vcc", 0);
smu->ClampCurrent("Vcc", 0.5);
smu->DriveVoltage("Vcc", 3.3);
smu->OnSmuPin("Vcc", true, false);
```

When this object is disposed, the communication session will be closed accordingly.

Core Library Function Mapping

1. AemDCPwr_InitChannels
2. AemDCPwr_Close

ReadRevision

Synopsis

long ReadRevision (moduleAlias, driverRevision, firmwareRevision)

Arguments

String ^ moduleAlias (in)

Specifies the alias of the selected module.

String ^ % driverRevision (out)

Returns the DLL version of AemDCPwr.dll and instruments DLL (eg. AM430.dll, AM471.dll etc). The DLL version is separated by a hyphen.

String ^ % firmwareRevision (out)

Returns firmware revision information for the device you are using. This argument returns both firmware version of DC and MC separated by a hyphen.

Core Library Function Mapping

1. AemDCPwr_ReadRevision

ReadCurrentTemperature

Synopsis

long ReadCurrentTemperature (moduleAlias, temperature)

Arguments

String ^ moduleAlias (in)

Specifies the alias of the selected module.

double ^ % temperature (out)

Returns the current onboard temperature, in degrees Celsius, of the SMU. This temperature sensor is placed between MC and DC cards.

Core Library Function Mapping

1. AemDCPwr_ReadCurrentTemperature

ReadAmbientTemperature

Synopsis

long ReadAmbientTemperature (moduleAlias, temperature)

Arguments

String ^ moduleAlias (in)

Specifies the alias of the selected module.

double ^ % temperature (out)

Returns the current temperature, read by GX FPGA in degrees Celsius. This temperature sensor is placed on bottom plane of MC card.

Core Library Function Mapping

1. AemDCPwr_ReadAmbientTemperature

ReadSerialNumber

Synopsis

long ReadSerialNumber (moduleAlias,serialNumber)

Arguments

String ^ moduleAlias (in)

Specifies the alias of the selected module.

String ^ % serialNumber (out)

Returns both master and daughter card serial numbers separated by a hyphen.

Core Library Function Mapping

1. AemDCPwr_ReadSerialNumber

Reset

Synopsis

long Reset (moduleAlias)

Arguments

String ^ moduleAlias (in)

Specifies the alias of the selected module.

Core Library Function Mapping

1. AemDCPwr_Reset

ResetChannel

Synopsis

long ResetChannel (moduleAlias, pinNumber)

Arguments

String ^ moduleAlias (in)

Specifies the alias of the selected module.

String ^ pinNumber (in)

Specifies the channel or channels to be acted upon. For example, "0" specifies channel 0 only. This can also be a channel list or a channel range.

A channel list is a comma (,) separated sequence of channel names (for example, "0,2" specifies channels 0 and 2).

A channel range is a lower bound channel followed by a hyphen (-) or colon (:) followed by an upper bound channel (for example, "0-2" specifies channels 0, 1, and 2).

Core Library Function Mapping

2. AemDCPwr_ResetChannel

GetError

Synopsis

long GetError (moduleAlias, code, bufferSize, description)

Arguments

String ^ moduleAlias (in)

Specifies the alias of the selected module.

int % code (out)

Returns the error code for the session or execution thread.

int bufferSize (in)

Specifies the number of bytes in the low level Char array specified for description.

If the error description, including the terminating NULL byte, contains more bytes than you indicate in this attribute, the API copies (buffer size - 1) bytes into the buffer, places an ASCII NULL byte at the end of the buffer, and returns the buffer size you must pass to get the entire value.

For example, if the value is ABCDEF and the bufferSize is 4, the API places ABC into the buffer and returns 7.

If you pass 0 for this attribute, you can pass VI_NULL for description.

String ^ % description (out)

Returns the error description for the error code. If there is no description, the API returns an empty string.

Core Library Function Mapping

1. AemDCPwr_GetError

GetErrorMessage

Synopsis

long GetErrorMessage (moduleAlias, errorCode, errorMessage)

Arguments

String ^ moduleAlias (in)

Specifies the alias of the selected module.

int % code (out)

Returns the error code for the session or execution thread.

int errorCode (in)

Specifies the error code.

String ^ % errorMessage (out)

Returns the error description for the error code. If there is no description, the API returns an empty string.

Core Library Function Mapping

1. AemDCPwr_GetErrorMessage

ClearError

Synopsis

long ClearError (moduleAlias)

Arguments

String ^ moduleAlias (in)

Specifies the alias of the selected module.

Core Library Function Mapping

1. AemDCPwr_ClearError

ConfigureMultiSiteMode

Synopsis

long ConfigureMultiSiteMode (moduleAlias, mode)

Arguments

String ^ moduleAlias (in)

Specifies the alias of the selected module.

int mode (in)

Specifies the operation mode:

mode	Operation
0	Single-site (Default)
1	Multi-site

Table 1 : Operation Mode

Core Library Function Mapping

1. AemDCPwr_ConfigureMultiSiteMode

QueryModuleType

Synopsis

long QueryModuleType (moduleAlias**,** module_type)

Arguments

String ^ moduleAlias (in)

Specifies the alias of the selected module.

Unsigned int% module_type (out)

Returns the module type of the specified instrument handle.

module_type	Module
0xA10A430e	AM430e
0xA10A471e	AM471e

Table 2 : ModuleType

Core Library Function Mapping

1. AemDCPwr_QueryModuleType

ConfigureOutputResistance

Synopsis

long ConfigureOutputResistance (pinAlias, resistance, Irange)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

double resistance (in)

Configures the value of the virtual resistance in series with FH.

Current Range	Programmable Resistance Range
1uA	±1MΩ
10uA	±100kΩ
100uA	±10kΩ
1mA	±1kΩ
10mA	±100Ω
100mA	±10Ω

Table 3: Current range

double Irange (in)

Specifies the targeted current range.

Core Library Function Mapping

1. AemDCPwr_ConfigureOutputResistance

ReadSlotAddress

Synopsis

long ReadSlotAddress (moduleAlias, slotAddress)

Arguments

String ^ moduleAlias (in)

Specifies the alias of the selected module.

Int% slotAddress (out)

Returns the slot number of the instrument where it is slotted in.

Core Library Function Mapping

1. ReadSlotAddress

ClampCurrent

Synopsis

long ClampCurrent (pinAlias, current)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

double current (in)

Specifies the clamp current level.

Core Library Function Mapping

1. AemDCPwr_ConfigureCurrentLimit

ClampCurrent

Synopsis

long ClampCurrent (pinAlias, current, iRange)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

double current (in)

Specifies the clamp current level.

double iRange (in)

Specifies the current range to be used.

Core Library Function Mapping

1. AemDCPwr_ConfigureCurrentLimitAndRange

ClampVoltage

Synopsis

long ClampVoltage (pinAlias, voltage)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

double voltage (in)

Specifies the clamp voltage level.

Core Library Function Mapping

1. AemDCPwr_ConfigureVoltageLimit

ClampVoltage

Synopsis

long ClampVoltage (pinAlias, voltage, vRange)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

double voltage (in)

Specifies the clamp voltagelevel.

double vRange (in)

Specifies the voltagerange to be used.

Core Library Function Mapping

1. AemDCPwr_ConfigureVoltageLimitAndRange

DriveCurrent

Synopsis

long DriveCurrent (pinAlias, current)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

double current (in)

Specifies the drive current level.

Core Library Function Mapping

1. AemDCPwr_ConfigureOutputFunction (function = 1)
2. AemDCPwr_ConfigureCurrentLevel

DriveCurrent

Synopsis

long DriveCurrent (pinAlias, current, iRange)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

double current (in)

Specifies the drive current level.

double iRange (in)

Specifies the current range to be used.

Core Library Function Mapping

1. AemDCPwr_ConfigureOutputFunction (function = 1)
2. AemDCPwr_ConfigureCurrentLevelAndRange

DriveVoltage

Synopsis

long DriveVoltage (pinAlias, voltage)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

double voltage (in)

Specifies the drive voltage level.

Core Library Function Mapping

1. AemDCPwr_ConfigureOutputFunction (function = 0)
2. AemDCPwr_ConfigureVoltageLevel

DriveVoltage

Synopsis

long DriveVoltage (pinAlias, voltage, vRange)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

double voltage (in)

Specifies the drive voltage level.

double vRange (in)

Specifies the voltage range to be used.

Core Library Function Mapping

1. AemDCPwr_ConfigureOutputFunction (function = 0)
2. AemDCPwr_ConfigureVoltageLevelAndRange

OffSmuPin

Synopsis

long OffSmuPin (pinAlias)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

Core Library Function Mapping

1. AemDCPwr_ConfigureOutputSwitch (switch = 0)

OnSmuPin

Synopsis

long OnSmuPin (pinAlias, remoteSenseEnabled, seriesResEnabled)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

bool remoteSenseEnabled (in)

Specifies if remote sense is enabled for the specified SMU pin.

bool seriesResEnabled (in)

Specifies if the built-in series resistor should be turned on at FORCE HIGH line.

Core Library Function Mapping

1. If seriesResEnabled is *true*:

   AemDCPwr_ConfigureSense (sense = 0)
   AemDCPwr_ConfigureOutputSwitch (switch = 2)

2. If seriesResEnabled is *false*:

   AemDCPwr_ConfigureSense (sense = 0 if remoteSenseEnabled = false)
   AemDCPwr_ConfigureOutputSwitch (switch = 1)

OnOffSmuPin

Synopsis

long OnOffSmuPin (pinSettings)

Arguments

array<AemSmuPinOnOff> ^ pinSettings (in)

An array of AemSmuPinOnOff structs that specifies which SMU pin to act upon with their respective on / off state.

Core Library Function Mapping

1. AemDCPwr_ConfigureOutputSwitch (switch = 2)

SetBandwidth

Synopsis

long SetBandwidth (pinAlias, bandwidth)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

int bandwidth (in)

Specifies the bandwidth setting.

Core Library Function Mapping

1. AemDCPwr_ConfigureOutputTransient

SetAIBandwidth

Synopsis

long SetAIBandwidth (pinAlias, mode, vRange, iRange, source_settling_time, clamp_settling_time, resistance, capacitance, bandwidth_setting)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

int mode (in)

Specifies whether the load is resistive or capacitive.

mode	Load
0	Resistor
1	Capacitor

Table 4: mode

double vRange (in)

Specifies the voltage level range, in volts.

double iRange (in)

Specifies the current level range, in amps.

double drive_settling_time (in)

Specifies the desired source settling time, in seconds.

double clamp_settling_time (in)

Specifies the desired settling time for compliance, in seconds.

double resistance (in)

Specifies the resistance of the resistor connected to the channel. This value will be ignored if the specified load is capacitor (mode = 1).

double capacitance (in)

Specifies the capacitance of the capacitor connected to the channel. This value will be ignored if the specified load is resistor (mode = 0).

double reserved (in)

This is a reserved argument. Set it to "0".

int bandwidth_setting (in)

Specifies the location of where the calculated bandwidth setting is placed in SetBandwidth. Available bandwidth_setting are 0 to 4.

Core Library Function Mapping

1. AemDCPwr_ComputeAIBandwidth

ConfigureVoltageLevel

Synopsis

long ConfigureVoltageLevel (pinAlias, level)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

double level (in)

Specifies voltage levels in Volts.

Core Library Function Mapping

1. AemDCPwr_ConfigureVoltageLevel

ConfigureVoltageLevelAndRange

Synopsis

long ConfigureVoltageLevelAndRange (pinAlias, level, range)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

double level (in)

Specifies voltage levels in Volts.

double range (in)

Specifies voltage level range in Volts.

Core Library Function Mapping

1. AemDCPwr_ConfigureVoltageLevelAndRange

ConfigureCurrentLimit

Synopsis

long ConfigureCurrentLimit (pinAlias, behavior, limit)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

double level (in)

Specifies voltage levels in Volts.

Core Library Function Mapping

1. AemDCPwr_ConfigureVoltageLevel

ConfigureCurrentLimitAndRange

Synopsis

long ConfigureCurrentLimitAndRange (pinAlias, behavior, limit, range)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

int behavior (in)

This is a reserved argument. Set it to "0".

double limit (in)

Specifies the current limit, in amps.

double range (in)

Specifies the current limit range, in amps.

Core Library Function Mapping

1. AemDCPwr_ ConfigureCurrentLimitAndRange

ConfigureCurrentLevel

Synopsis

long ConfigureCurrentLevel (pinAlias, level)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

double level (in)

Specifies the current level, in amps.

Core Library Function Mapping

1. AemDCPwr_ConfigureCurrentLevel

ConfigureCurrentLevelAndRange

Synopsis

long ConfigureCurrentLevelAndRange (pinAlias, level, range)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

double level (in)

Specifies the current level, in amps.

double range (in)

Specifies the current level range, in amps.

Core Library Function Mapping

1. AemDCPwr_ConfigureCurrentLevelAndRange

ConfigureVoltageLimit

Synopsis

long ConfigureVoltageLimit (pinAlias, limit)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

double limit (in)

Specifies the voltage limit, in volts.

Core Library Function Mapping

2. AemDCPwr_ConfigureVoltageLimit

ConfigureVoltageLimitAndRange

Synopsis

long ConfigureVoltageLimitAndRange (pinAlias, limit, range)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

double limit (in)

Specifies the voltage limit, in volts.

double range (in)

Specifies the voltage limit range, in volts.

Core Library Function Mapping

1. AemDCPwr_ ConfigureVoltageLimitAndRange

ConfigureOutputFunction

Synopsis

long ConfigureOutputFunction (pinAlias, function)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

int function (in)

Select DVCI or DICV mode.

function	Setting
DVCI	0 (Default)
DICV	1

Table 5 : Output Function

Core Library Function Mapping

1. AemDCPwr_ConfigureOutputFunction

ConfigureOutputSwitch

Synopsis

long ConfigureOutputFunction (pinAlias, switch)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

int switch (in)

Configure the output state of the selected channels.

switch	Description
0	Switch off the channel
1	Switch on the channel
2	Switch on the channel, and insert a 100K resistor at Force-High output

Table 6:Output Switch

Core Library Function Mapping

1. AemDCPwr_ ConfigureOutputSwitch

ConfigureOutputEnabled

Synopsis

long ConfigureOutputEnabled (pinAlias, enabled)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

bool enabled (in)

Specifies whether to enable (true) or disable (false) SMU feedback control loop.

Core Library Function Mapping

1. AemDCPwr_ConfigureOutputEnabled

ConfigureOutputTransient

Synopsis

long ConfigureOutputTransient (pinAlias, transcient)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

int transcient (in)

Specifies the transient (bandwidth) setting of the specified channel.

transient	Description
0	Slow
1	Normal (Default)
2	Fast
3	Custom bandwidth store location 0
4	Custom bandwidth store location 1
5	Custom bandwidth store location 2
6	Custom bandwidth store location 3
7	Custom bandwidth store location 4

Table 7: Output Function

Core Library Function Mapping

1. AemDCPwr_ConfigureOutputTransient

ComputeAIBandwidth

Synopsis

long ComputeAIBandwidth (pinAlias, mode, vRange, iRange, driveSettlingTime, clampSettlingTime, resistance, capacitance, reserved, storeLocation)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

int mode (in)

Specifies whether the load is resistive or capacitive.

mode	Load
0	Resistor
1	Capacitor

Table 8: Mode

double vRange (in)

Specifies the voltage level range, in volts.

double iRange (in)

Specifies the current level range, in amps.

double driveSettlingTime (in)

Specifies the desired source settling time, in seconds.

double clampSettlingTime (in)

Specifies the desired settling time for compliance, in seconds.

double resistance (in)

Specifies the resistance of the resistor connected to the channel. This value will be ignored if the specified load is capacitor (mode = 1).

double capacitance (in)

Specifies the capacitance of the capacitor connected to the channel. This value will be ignored if the specified load is resistor (mode = 0).

double reserved (in)

This is a reserved argument. Set it to "0".

int storeLocation (in)

Specifies the location of where the calculated bandwidth setting is placed in ConfigureOutputTransient. Available store_location are 0 to 4.

Core Library Function Mapping

1. AemDCPwr_ConfigureOutputTransient

ConfigurePulse

Synopsis

long ConfigurePulse (pinAlias, function, range, base, pulse, pulse_s, hold_s, reserved, measurePercentage, cycles)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

int function (in)

Select DVCI or DICV mode.

function	Description
0	The channel maintains a constant voltage by adjusting the current (DVCI)
1	The channel maintains a constant current by adjusting the voltage (DICV)

Table 9: Output Function

double range (in)

Range is vRange if DVCI, iRange if DICV.

double base (in)

Specifies the base value of the pulse.

double pulse (in)

Specifies the pulse value of the pulse.

double pulse_s (in)

Specifies the pulse width in second

0<= pulse_s<=20ms

double hold_s (in)

Specifies the period of the base value in second.

100us <= hold_s <= 500ms

int reserved (in)

This is a reserved argument. Set it to "0".

double measurePercentage (in)

Specifies the time for measurement during drive pulse.

int cycles (in)

Specifies the number of pulses targeted.

1<= cycles <= 64

Core Library Function Mapping

1. AemDCPwr_ConfigurePulse

ConfigureContinuousPulse

Synopsis

long ConfigureContinuousPulse (pinAlias, function, base, pulse, range, pulse_s, hold_s, cycle, infiniteCycle)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

int function (in)

Select DVCI or DICV mode.

function	Description
0	The channel maintains a constant voltage by adjusting the current (DVCI)
1	The channel maintains a constant current by adjusting the voltage (DICV)

Table 10: OutputFunction

double base (in)

Specifies the base value of the pulse.

double pulse (in)

Specifies the pulse value of the pulse.

double range (in)

Range is vRange if DVCI, iRange if DICV.

double pulse_s (in)

Specifies the pulse width in second

0<= pulse_s<=20ms

double hold_s (in)

Specifies the period of the base value in second.

100us <= hold_s <= 500ms

int reserved (in)

This is a reserved argument. Set it to "0".

int cycles (in)

Specifies the number of pulses targeted.

1<= cycles <= 64

int infiniteCycle (in)

Specifies the number of pulses cycle.

infiniteCycle	Description
0	The number of pulse is referred to argument cycle.
1	Infinite generation of pulse.

Table 11: Infinite Cyele Selection

Core Library Function Mapping

1. AemDCPwr_ConfigureContinuousPulse

StartContinuousPulse

Synopsis

long StartContinuousPulse (pinAlias)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

Core Library Function Mapping

1. AemDCPwr_StartContinuousPulse

StopContinuousPulse

Synopsis

long StopContinuousPulse (pinAlias)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

Core Library Function Mapping

1. AemDCPwr_StopContinuousPulse

ConfigureExtenderModule

Synopsis

long ConfigureExtenderModule (moduleAlias, moduleStatus)

Arguments

String ^ moduleAlias (in)

Specifies the instrument handle.

int % moduleStatus (out)

Returns the status of the extender module.

moduleStatus	Description
0	Extender card not detected and power supply generation is disabled.
1	Extender card is detected and power supply generation is enabled.

Table 12: Module Status

Core Library Function Mapping

1. AemDCPwr_ConfigureExtenderModule

ConfigureAcquireArrayBWLimit

Synopsis

long ConfigureAcquireArrayBWLimit (pinAlias, setting)

Arguments

String ^ pinAlias (in)

Specifies the instrument handle.

int setting (in)

Configures the bandwidth setting for array measurement.

Core Library Function Mapping

1. AemDCPwr_ConfigureAcquireArrayBWLimit

SetNPLC

Synopsis

long SetNPLC (pinAlias, nplc)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

double nplc (in)

Specifies the NPLC setting.

Core Library Function Mapping

1. AemDCPwr_ConfigureSamplingTime (units = 1)

ConfigureAcquireRecordLength

Synopsis

long ConfigureAcquireRecordLength (pinAlias, length)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

int length (in)

Specifies the number of samples to be acquired by AcquireMultiple.

Core Library Function Mapping

1. AemDCPwr_ConfigureAcquireRecordLength

SetIntegration

Synopsis

long SetIntegration (pinAlias, integration)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

SmuIntegration integration (in)

Aemulus.Hardware.SmuIntegration that represents the integration setting.

Core Library Function Mapping

1. AemDCPwr_ConfigurePLF

ReadCurrent

Synopsis

long ReadCurrent (pinAlias, current)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

double * current (out)

Returns the measured current.

Core Library Function Mapping

1. AemDCPwr_ConfigureMeasureMode (mode = 2)
2. AemDCPwr_Measure (measurementType = 0)

ReadCurrentWithAverage

Synopsis

long ReadCurrentWithAverage (pinAlias, sampleSize, average, rawSamples)

Arguments

String ^ pinAlias (in)
Specifies the alias of the target SMU pin.

unsigned int sampleSize (in)
Specifies the number of samples to be taken for averaging.

double * average (out)
Returns the averaged measured current.

array<double> ^ % rawSamples (out)
Returns an array of measured current.

Core Library Function Mapping

AemDCPwr_ConfigureMeasureMode (mode = 2)
AemDCPwr_MeasureArray (measurementType = 0)

ReadVoltage

Synopsis

long ReadVoltage (pinAlias, voltage)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

double * voltage (out)

Returns the measured voltage.

Core Library Function Mapping

1. AemDCPwr_ConfigureMeasureMode (mode = 2)
2. AemDCPwr_Measure (measurementType = 1)

ReadVoltageWithAverage

Synopsis

long ReadVoltageWithAverage (pinAlias, sampleSize, average, rawSamples)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

unsigned int sampleSize (in)

Specifies the number of samples to be taken for averaging.

double * average (out)

Returns the averaged measured voltage.

double* measurement (out)

Returns an array of measured voltage.

Core Library Function Mapping

1. AemDCPwr_ConfigureMeasureMode (mode = 2)
2. AemDCPwr_MeasureArray (measurementType = 1)

ReadVoltageAndCurrent

Synopsis

long ReadVoltageAndCurrent (pinAlias, voltage, current)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

array<double> ^ % voltage (out)

Returns the measured voltage.

array<double> ^ % current (out)

Returns the measured current.

Core Library Function Mapping

1. AemDCPwr_ConfigureMeasureMode (mode = 2)
2. AemDCPwr_Measure (measurementType = 1)
3. AemDCPwr_MeasureArray (measurementType = 0)

KelvinContactCheck

Synopsis

long KelvinContactCheck (pinAlias, HI_h_LO_l, threshold_mohm, delay_s, pass_h_fail_l, mohm)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

int shi_h_slo_l (in)

Specifies whether to perform contact check on Force-High/Sense-High pair or Force-Low/Sense-Low pair.
Set to "1" for Force-High/Sense-High pair.
Set to "0" for Force-Low/Sense-Low pair.

int threshold_mohm (in)

Specifies the threshold to determine pass or fail, in ohms.

double delay_s (in)

Specifies measurement delay (the delay after drive current and before read voltage).

int* pass_h_fail_l (out)

Returns test result based on threshold_mohm. "1" means pass, "0" means fail.

int* mohm (out)

Returns the measured resistance, in ohms.

Core Library Function Mapping

1. AemDCPwr_ContactCheck

ConfigureSamplingTime

Synopsis

long ConfigureSamplingTime (pinAlias, samplingTime, unit)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

double samplingTime (in)

Specifies the sampling time.

int unit (in)

Specifies the unit of the sampling time:

units	Description
0	Seconds
1	PLC

Table 13: Sampling Time Unit

Core Library Function Mapping

1. AemDCPwr_ ConfigureSamplingTime

ConfigurePLF

Synopsis

long ConfigurePLF (pinAlias, plf)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

double plf (in)

Specifies the power line frequency.

plf	Description
50.0	50Hz power line frequency (Default)
60.0	60Hz power line frequency

Table 14: Power Line Frequency

Core Library Function Mapping

1. AemDCPwr_ConfigurePLF

ConfigureSense

Synopsis

long ConfigureSense (pinAlias**,** sense)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

double sense (in)

Specifies local or remote sense operation.

Core Library Function Mapping

1. AemDCPwr_ConfigureSense

Measure

Synopsis

long Measure (pinAlias, printToTxt, measurementType, measurement)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

int measurementType (in)

Specifies voltage or current measurement.

double % measurement (out)

Returns the measured results.

Core Library Function Mapping

1. AemDCPwr_Measure

MeasureArray

Synopsis

long MeasureArray (pinAlias, printToTxt, measurementType, measurement)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

bool printToTxt (in)

Specifies whether to print out the raw data in a text document.

int measurementType (in)

Specifies the type of measurement.

measurementType	Description
0	Measure Current
1	Measure Voltage
2	Measure both current and voltage

Table 15: Measurement Types

array <double> ^ % measurement (in)

Returns an array of measured results.

Core Library Function Mapping

1. AemDCPwr_MeasureArray

MeasureMultiple

Synopsis

long MeasureMultiple (pinAlias, voltageMeasurements, currentMeasurements)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

Array <double> ^ % voltageMeasurements (out)

Space allocated by the caller. Returns an array of voltage measurements. The measurements in the array are returned in the same order as the channels specified in channelName argument.

Array <double> ^ % currentMeasurements (out)

Space allocated by the caller. Returns an array of current measurements. The measurements in the array are returned in the same order as the channels specified in channelName argument.

Core Library Function Mapping

1. AemDCPwr_ MeasureMultiple

ClearAcquireRecordLength

Synopsis

long ClearAcquireRecordLength (pinAlias)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

Core Library Function Mapping

1. AemDCPwr_ClearAcquireRecordLength

ConfigureMeasureMode

Synopsis

long ConfigureMeasureMode (pinAlias, mode)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

int mode (in)

Specifies measurement operation mode.

mode	Description
0	AemDCPwr_AcquireMultiple starts immediately after AemDCPwr_ConfigureVoltageLevel, AemDCPwr_ConfigureVoltageLevelAndRange, AemDCPwr_ConfigureCurrentLevel or AemDCPwr_ConfigureCurrentLevelAndRange functions. It will not wait until the channel output reach the desired voltage or current level.
1	AemDCPwr_AcquireMultiple starts after receive the trigger signal. The selection of trigger signal is based on input setting of AemDCPwr_ConfigureInputTriggerSelect function.
2	AemDCPwr_AcquireMultiple starts upon being called. Use this for
3	AemDCPwr_AcquireArray starts after receiving the trigger signal. The selection of trigger signal is based on input setting of AemDCPwr_ConfigureInputTriggerSelect function.

Core Library Function Mapping

1. AemDCPwr_ConfigureMeasureMode

AcquireMultiple

Synopsis

long AcquireMultiple (pinAlias, timeOut, count, voltMeas, currMeas, incompliance, actualCount)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

double timeOut (in)

Specifies the maximum time allowed for this function to complete, in seconds. If the function does not complete within this time interval, an error will be returned.

int count (in)

Specifies the number of measurements to fetch.

array<double> ^ % voltMeas (out)

Space allocated by the caller. Returns an array of voltage measurement.

array<double> ^ % currMeas (out)

Space allocated by the caller. Returns an array of current measurement.

array<unsigned short> ^ % inCompliance (out)

Space allocated by the caller. Returns an array of boolean values indicating whether the output was in compliance at the time the measurement was taken.

int % actualCount (out)

Indicates the number of measured values actually retrieved from the instrument.

Core Library Function Mapping

1. AemDCPwr_AcquireMultiple

AcquireArray

Synopsis

long AcquireArray (pinAlias, timeOut, count, voltageMeasurements, currentMeasurements, actualCount)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

double timeOut (in)

Specifies the maximum time allowed for this function to complete, in seconds. If the function does not complete within this time interval, an error will be returned.

int count (in)

Specifies the number of measurements to fetch.

array<double> ^ % voltageMeasurements (out)

Space allocated by the caller. Returns an array of voltage measurement.

array<double> ^ % currentMeasurements (out)

Space allocated by the caller. Returns an array of current measurement.

int % actualCount (out)

Indicates the number of measured values actually retrieved from the instrument.

Core Library Function Mapping

1. AemDCPwr_AcquireArray

MeasurePulseVI

Synopsis

long MeasurePulseVI (pinAlias, cycles, vranges, iranges, voltages, currents)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

Int% cycles (out)

Returns the number of targeted cycle when sourcing pulse.

double % vranges (out)

Returns the voltage range when DVCI.

double % iranges (out)

Returns the current range when DICV.

array <double> ^ % voltages (out)

Return an array of the voltage measurements during ConfigurePulse.

array <double> ^ % currents (out)

Return an array of the current measurements during ConfigurePulse.

Core Library Function Mapping

1. AemDCPwr_MeasurePulseVI

ConfigureAcquireArrayInterval

Synopsis

long ConfigureAcquireArrayInterval (pinAlias, interval_s)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

double interval_s (in)

By default, upon power-up interval_s is 1us. Resolution of interval_s is 1us. Unit of interval_s is in seconds and therefore automatically rounded to the closest us. Minimum interval_s is 1us and maximum interval_s is 1ms.

Core Library Function Mapping

1. AemDCPwr_ConfigureAcquireArrayInterval

Section 4: Query

GetIntegration

Synopsis

long GetIntegration (pinAlias, integration)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

SmuIntegration * integration (out)

Aemulus.Hardware.SmuIntegration that represents the integration setting.

GetNPLC

Synopsis

long GetNPLC (pinAlias, nplc)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

double * nplc (out)

Returns the NPLC (Number of Power Line Cycle) setting of specified SMU pin.

ReadVoltageLevelRange

Synopsis

long ReadVoltageLevelRange (pinAlias, range)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

Double % range (out)

Returns the voltage level range of the specified channel, in volts.

Core Library Function Mapping

1. AemDCPwr_ReadVoltageLevelRange

ReadVoltageLimitRange

Synopsis

long ReadVoltageLimitRange (pinAlias, range)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

Double % range (out)

Returns the voltage limit range of the specified channel, in volts.

Core Library Function Mapping

1. AemDCPwr_ReadVoltageLimitRange

ReadCurrentLevelRange

Synopsis

long ReadCurrentLevelRange (pinAlias, range)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

Double % range (out)

Returns the curent level range of the specified channel, in amps.

Core Library Function Mapping

1. AemDCPwr_ReadVoltageLimitRange

ReadCurrentLimitRange

Synopsis

long ReadCurrentLimitRange (pinAlias, range)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

Double % range (out)

Returns the current limit range of the specified channel, in amps.

Core Library Function Mapping

1. AemDCPwr_ ReadCurrentLimitRange

QueryInCompliance

Synopsis

long QueryInCompliance (pinAlias, inCompliance)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

bool % inCompliance (in)

Returns the compliance state of the specified channels.

Core Library Function Mapping

1. AemDCPwr_QueryInCompliance

QueryOutputState

Synopsis

long QueryOutputState (pinAlias, outputState, inState)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

int outputState (in)

Specifies the state of the specified channels:

outputState	Description
0	The channel maintains a constant voltage by adjusting the current (DVCI)
1	The channel maintains a constant current by adjusting the voltage (DICV)

Table 16:Output States

bool % inState (out)

Returns whether the device output channel is in the specified state.

Core Library Function Mapping

1. AemDCPwr_QueryOutputState

QueryMaxVoltageLevel

Synopsis

long QueryMaxVoltageLevel (pinAlias, currentLimit, maxVoltageLevel)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

double currentLimit (in)

Specifies the current limit to use when calculating the maxVoltageLevel.

double % maxVoltageLevel (out)

Returns the maximum voltage level that can be achieved given the currentLimit.

Core Library Function Mapping

1. AemDCPwr_QueryMaxVoltageLevel

QueryMaxCurrentLimit

Synopsis

long QueryMaxCurrentLimit (pinAlias, voltageLevel, maxCurrentLimit)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

double voltageLevel (in)

Specifies the voltage level to use when calculating the maxCurrentLimit.

double % maxCurrentLimit (out)

Returns the maximum current limit that can be achieved given the voltageLevel.

Core Library Function Mapping

1. AemDCPwr_QueryMaxCurrentLimit

QueryMinCurrentLimit

Synopsis

long QueryMinCurrentLimit (pinAlias, voltageLevel, minCurrentLimit)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

double voltageLevel (in)

Specifies the voltage limit to use when calculating the minCurrentLimit.

double % minCurrentLimit (out)

Returns the minimum current level that can be achieved given the voltageLevel.

Core Library Function Mapping

1. AemDCPwr_QueryMinCurrentLimit

QueryAcquireRecordLength

Synopsis

long QueryAcquireRecordLength (pinAlias, length)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

int length (in)

Returns the acquired record length by AemDCPwr_AquireMultiple.

Core Library Function Mapping

1. AemDCPwr_QueryAcquireRecordLength

AcquireArray

Synopsis

long AcquireArray (pinAlias, timeout, voltageMeasurements, currentMeasurements, actualCount)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

double timeout (in)

Specifies the maximum time allowed for this function to complete, in seconds. If the function does not complete within this time interval, an error will be returned.

double % voltageMeasurements (out)

Space allocated by the caller. Returns an array of voltage measurement.

double % currentMeasurements (out)

Space allocated by the caller. Returns an array of current measurement.

int % actualCount (out)

Indicates the number of measured values actually retrieved from the instrument.

Core Library Function Mapping

1. AemDCPwr_AcquireArray

ExtTrigArm_ReadCurrentArray

Synopsis

long ExtTrigArm_ReadCurrentArray (pinAlias, trigMode, meas_delay_after_trig, sampleSize, sample_to_sample_delay, timeout)

Arguments

String ^ pinAlias (in)
Specifies the alias of the target SMU pin.

int trigMode (in)
Specifies the triggering mode.

double meas_delay_after_trig (in)
Specifies the measure delay after being triggered, in seconds.

int sampleSize (in)
Specifies the number of samples to be taken.

double sample_to_sample_delay (in)
Specifies the sample to sample interval delay, in seconds.

double timeout (in)
Specifies the trigger arm timeout, in second.

Descriptions

This function arms the specified SMU pin to prepare for current measurement in an array upon triggered at the external trigger connection.

Core Library Function Mapping

1. AemDCPwr_MapTriggerInToTriggerOut (inputTerminal = 0, outputTerminal =23)

2. AemDCPwr_ConfigureInputTriggerSelect (triggerInput = 23)
3. AemDCPwr_ConfigureTriggerEdgeLevel (triggerEnum = 23)
4. AemDCPwr_ConfigureMeasureMode (mode = 3)

ExtTrigArm_ReadCurrentArray

Synopsis

long ExtTrigArm_ReadCurrentArray (pinAlias, trigMode, meas_delay_after_trig, sampleSize, sample_to_sample_delay, timeout, ignore_trigger_count)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

int trigMode (in)

Specifies the triggering mode.

double meas_delay_after_trig (in)

Specifies the measure delay after being triggered, in seconds.

int sampleSize (in)

Specifies the number of samples to be taken.

double sample_to_sample_delay (in)

Specifies the sample to sample interval delay, in seconds.

double timeout (in)

Specifies the trigger arm timeout, in second.

int ignore_trigger_count (in)

Specifies the number of triggers to be ignored, before performing measurement.
If ignore_trigger_count = 0, first trigger is a valid trigger.
If ignore_trigger_count = 2, module will ignore the first two triggers and only begin measuring after third trigger.

Descriptions

This function arms the specified SMU pin to prepare for current measurement in an array upon triggered at the external trigger connection.

1. AemDCPwr_MapTriggerInToTriggerOut (inputTerminal = 0, outputTerminal =23)
2. AemDCPwr_ConfigureInputTriggerSelect (triggerInput = 23)
3. AemDCPwr_ConfigureTriggerEdgeLevelExtra (triggerEnum = 23)
4. AemDCPwr_ConfigureMeasureMode (mode = 3)

ExtTrigArm_ReadCurrentArray

Synopsis

long ExtTrigArm_ReadCurrentArray (pinAlias, trigMode, meas_delay_after_trig, sampleSize, sample_to_sample_delay, ignore_trigger_count)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

int trigMode (in)

Specifies the triggering mode.

double meas_delay_after_trig (in)

Specifies the measure delay after being triggered, in seconds.

int sampleSize (in)

Specifies the number of samples to be taken.

double sample_to_sample_delay (in)

By default, upon power-up interval_s is 1us. Resolution of interval_s is 1us. Unit of interval_s is in seconds and therefore automatically rounded to the closest us. Minimum interval_s is 1us and maximum interval_s is 1ms.

int ignore_trigger_count (in)

Specifies the number of triggers to be ignored, before performing measurement.
If ignore_trigger_count = 0, first trigger is a valid trigger.
If ignore_trigger_count = 2, module will ignore the first two triggers and only begin measuring after third trigger.

Descriptions

This function arms the specified SMU pin to prepare for current measurement in an array upon triggered at the external trigger connection. This function calculates the NPLC according to the sampleSize and sample_to_sample_delay before arms the SMU pin.

Core Library Function Mapping

1. AemDCPwr_MapTriggerInToTriggerOut (inputTerminal = 0, outputTerminal =23)
2. AemDCPwr_ConfigureSamplingTime
3. AemDCPwr_ConfigureAcquireArrayInterval
4. AemDCPwr_ConfigureInputTriggerSelect (triggerInput = 23)
5. AemDCPwr_ConfigureTriggerEdgeLevel(for module AM430e) or AemDCPwr_ConfigureTriggerEdgeLevelExtra for other module (triggerEnum = 23)
6. AemDCPwr_ConfigureMeasureMode

ExtTrigGet_ReadCurrentArray

Synopsis

long ExtTrigGet_ReadCurrentArray (pinAlias, sampleSize, rawSamples, min, max, average)

Arguments

String ^ pinAlias (in)
Specifies the alias of the target SMU pin.

int * sampleSize (out)
Returns the number of samples captured.

double * rawSamples (out)
Returns an array of sampled current.

double* min (out)
Returns the minimum value of the array.

double* max (out)
Returns the maximum value of the array.

double* average (out)
Returns the average value of the array.

Descriptions

This function retrieves the measured current when measurement is completed.

Core Library Function Mapping

1. AemDCPwr_AcquireCurrentArray

ExtTrigArm_Release

Synopsis

long ExtTrigArm_Release (pinAlias)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

Descriptions

This function clears the trigger status.

Core Library Function Mapping

1. AemDCPwr_ClearAcquireRecordLength
2. AemDCPwr_ConfigureInputTriggerSelect (triggerInput = 0)

ExtTrigArm_ReadVoltageArray

Synopsis

long ExtTrigArm_ReadVoltageArray (pinAlias, trigMode, meas_delay_after_trig, sampleSize, sample_to_sample_delay, timeout)

Arguments

String ^ pinAlias (in)
: Specifies the alias of the target SMU pin.

int trigMode (in)
: Specifies the triggering mode.

double meas_delay_after_trig (in)
: Specifies the measure delay after being triggered, in seconds.

int sampleSize (in)
: Specifies the number of samples to be taken.

double sample_to_sample_delay (in)
: Specifies the sample to sample interval delay, in seconds.

double timeout (in)
: Specifies the trigger arm timeout, in second.

Descriptions

This function arms the specified SMU pin to prepare for voltage measurement in an array upon triggered at the external trigger connection.

Core Library Function Mapping

1. AemDCPwr_MapTriggerInToTriggerOut (inputTerminal = 0, outputTerminal =23)
2. AemDCPwr_ConfigureInputTriggerSelect (triggerInput = 23)
3. AemDCPwr_ConfigureTriggerEdgeLevel (triggerEnum = 23)

4. AemDCPwr_ConfigureMeasureMode (mode = 3)

ExtTrigArm_ReadVoltageArray

Synopsis

long ExtTrigArm_ReadVoltageArray (pinAlias, trigMode, meas_delay_after_trig, sampleSize, sample_to_sample_delay, timeout, ignore_trigger_count)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

int trigMode (in)

Specifies the triggering mode.

double meas_delay_after_trig (in)

Specifies the measure delay after being triggered, in seconds.

int sampleSize (in)

Specifies the number of samples to be taken.

double sample_to_sample_delay (in)

Specifies the sample to sample interval delay, in seconds.

double timeout (in)

Specifies the trigger arm timeout, in second.

int ignore_trigger_count (in)

Specifies the number of triggers to be ignored, before performing measurement.
If ignore_trigger_count = 0, first trigger is a valid trigger.
If ignore_trigger_count = 2, module will ignore the first two triggers and only begin measuring after third trigger.

Descriptions

This function arms the specified SMU pin to prepare for voltage measurement in an array upon triggered at the external trigger connection.

1. AemDCPwr_MapTriggerInToTriggerOut (inputTerminal = 0, outputTerminal =23)
2. AemDCPwr_ConfigureInputTriggerSelect (triggerInput = 23)
3. AemDCPwr_ConfigureTriggerEdgeLevel Extra(triggerEnum = 23)
4. AemDCPwr_ConfigureMeasureMode (mode = 3)

ExtTrigArm_ReadVoltageArray

Synopsis

long ExtTrigArm_ReadVoltageArray (pinAlias, trigMode, meas_delay_after_trig, sampleSize, sample_to_sample_delay, ignore_trigger_count)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

int trigMode (in)

Specifies the triggering mode.

double meas_delay_after_trig (in)

Specifies the measure delay after being triggered, in seconds.

int sampleSize (in)

Specifies the number of samples to be taken.

double sample_to_sample_delay (in)

By default, upon power-up interval_s is 1us. Resolution of interval_s is 1us. Unit of interval_s is in seconds and therefore automatically rounded to the closest us. Minimum interval_s is 1us and maximum interval_s is 1ms.

int ignore_trigger_count (in)

Specifies the number of triggers to be ignored, before performing measurement.
If ignore_trigger_count = 0, first trigger is a valid trigger.
If ignore_trigger_count = 2, module will ignore the first two triggers and only begin measuring after third trigger.

Descriptions

This function arms the specified SMU pin to prepare for voltage measurement in an array upon triggered at the external trigger connection. This function calculates the NPLC according to the sampleSize and sample_to_sample_delay before arms the SMU pin.

1. AemDCPwr_MapTriggerInToTriggerOut (inputTerminal = 0, outputTerminal =23)
2. AemDCPwr_ConfigureSamplingTime
3. AemDCPwr_ConfigureAcquireArrayInterval
4. AemDCPwr_ConfigureInputTriggerSelect (triggerInput = 23)
5. AemDCPwr_ConfigureTriggerEdgeLevel(for module AM430e) or AemDCPwr_ConfigureTriggerEdgeLevelExtra for other module (triggerEnum = 23)
6. AemDCPwr_ConfigureMeasureMode (mode = 3)

ExtTrigGet_ReadVoltageArray

Synopsis

long ExtTrigGet_ReadVoltageArray (pinAlias, sampleSize, rawSamples, min, max, average)

Arguments

String ^ pinAlias (in)
: Specifies the alias of the target SMU pin.

int * sampleSize (out)
: Returns the number of samples captured.

double * rawSamples (out)
: Returns an array of sampled voltage.

double* min (out)
: Returns the minimum value of the array.

double* max (out)
: Returns the maximum value of the array.

double* average (out)
: Returns the average value of the array.

Descriptions

This function retrieves the measured voltage when measurement is completed.

Core Library Function Mapping

1. AemDCPwr_AcquireArray

TrigArm_ReadCurrentArray

Synopsis

long TrigArm_ReadCurrentArray (pinAlias, trigSource, trigMode, meas_delay_after_trig, sampleSize, sample_to_sample_delay, timeout)

Arguments

String ^ pinAlias (in)
: Specifies the alias of the target SMU pin.

int trigSource (in)
: Specifies the triggering source.

int trigMode (in)
: Specifies the triggering mode.

double meas_delay_after_trig (in)
: Specifies the measure delay after being triggered, in seconds.

int sampleSize (in)
: Specifies the number of samples to be taken.

double sample_to_sample_delay (in)
: Specifies the sample to sample interval delay, in seconds.

double timeout (in)
: Specifies the trigger arm timeout, in second.

Descriptions

This function arms the specified SMU pin to prepare for current measurement in an array upon triggered.

1. AemDCPwr_ConfigureInputTriggerSelect
2. AemDCPwr_ConfigureTriggerEdgeLevel
3. AemDCPwr_ConfigureMeasureMode (mode = 3)

TrigArm_ReadCurrentArray

Synopsis

long TrigArm_ReadCurrentArray (pinAlias, trigSource, trigMode, meas_delay_after_trig, sampleSize, sample_to_sample_delay, timeout, ignore_trigger_count)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

int trigSource (in)

Specifies the triggering source.

int trigMode (in)

Specifies the triggering mode.

double meas_delay_after_trig (in)

Specifies the measure delay after being triggered, in seconds.

int sampleSize (in)

Specifies the number of samples to be taken.

double sample_to_sample_delay (in)

Specifies the sample to sample interval delay, in seconds.

double timeout (in)

Specifies the trigger arm timeout, in second.

int ignore_trigger_count (in)

Specifies the number of triggers to be ignored, before performing measurement.
If ignore_trigger_count = 0, first trigger is a valid trigger.
If ignore_trigger_count = 2, module will ignore the first two triggers and only begin measuring after third trigger.

Descriptions

This function arms the specified SMU pin to prepare for current measurement in an array upon triggered.

Core Library Function Mapping

1. AemDCPwr_ConfigureInputTriggerSelect
2. AemDCPwr_ConfigureTriggerEdgeLevelExtra
3. AemDCPwr_ConfigureMeasureMode (mode = 3)

TrigArm_ReadCurrentArray

Synopsis

long TrigArm_ReadCurrentArray (pinAlias, trigSource, trigMode, meas_delay_after_trig, sampleSize, sample_to_sample_delay, ignore_trigger_count)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

int trigSource (in)

Specifies the triggering source.

int trigMode (in)

Specifies the triggering mode.

double meas_delay_after_trig (in)

Specifies the measure delay after being triggered, in seconds.

int sampleSize (in)

Specifies the number of samples to be taken.

double sample_to_sample_delay (in)

By default, upon power-up interval_s is 1us. Resolution of interval_s is 1us. Unit of interval_s is in seconds and therefore automatically rounded to the closest us. Minimum interval_s is 1us and maximum interval_s is 1ms.

int ignore_trigger_count (in)

Specifies the number of triggers to be ignored, before performing measurement.
If ignore_trigger_count = 0, first trigger is a valid trigger.
If ignore_trigger_count = 2, module will ignore the first two triggers and only begin measuring after third trigger.

Descriptions

This function arms the specified SMU pin to prepare for current measurement in an array upon triggered. This function calculates the NPLC according to the sampleSize and sample_to_sample_delay before arms the SMU pin.

Core Library Function Mapping

1. AemDCPwr_ConfigureSamplingTime
2. AemDCPwr_ConfigureAcquireArrayInterval
3. AemDCPwr_ConfigureInputTriggerSelect
4. AemDCPwr_ConfigureTriggerEdgeLevel(for module AM430e) or AemDCPwr_ConfigureTriggerEdgeLevelExtra for other module
5. AemDCPwr_ConfigureMeasureMode (mode = 3)

TrigGet_ReadCurrentArray

Synopsis

long TrigGet_ReadCurrentArray (pinAlias, sampleSize, rawSamples, min, max, average)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

int * sampleSize (out)

Returns the number of samples captured.

double * rawSamples (out)

Returns an array of sampled current.

double* min (out)

Returns the minimum value of the array.

double* max (out)

Returns the maximum value of the array.

double* average (out)

Returns the average value of the array.

Descriptions

This function retrieves the measured current when measurement is completed.

Core Library Function Mapping

2. AemDCPwr_AcquireCurrentArray

TrigArm_ReadVoltageArray

Synopsis

long TrigArm_ReadVoltageArray (pinAlias, trigSource, trigMode, meas_delay_after_trig, sampleSize, sample_to_sample_delay, timeout)

Arguments

String ^ pinAlias (in)
: Specifies the alias of the target SMU pin.

int trigSource (in)
: Specifies the triggering source.

int trigMode (in)
: Specifies the triggering mode.

double meas_delay_after_trig (in)
: Specifies the measure delay after being triggered, in seconds.

int sampleSize (in)
: Specifies the number of samples to be taken.

double sample_to_sample_delay (in)
: Specifies the sample to sample interval delay, in seconds.

double timeout (in)
: Specifies the trigger arm timeout, in second.

Descriptions

This function arms the specified SMU pin to prepare for voltage measurement in an array upon triggered.

1. AemDCPwr_ConfigureInputTriggerSelect
2. AemDCPwr_ConfigureTriggerEdgeLevel
3. AemDCPwr_ConfigureMeasureMode (mode = 3)

TrigArm_ReadVoltageArray

Synopsis

long TrigArm_ReadVoltageArray (pinAlias, trigSource, trigMode, meas_delay_after_trig, sampleSize, sample_to_sample_delay, timeout, ignore_trigger_count)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

int trigSource (in)

Specifies the triggering source.

int trigMode (in)

Specifies the triggering mode.

double meas_delay_after_trig (in)

Specifies the measure delay after being triggered, in seconds.

int sampleSize (in)

Specifies the number of samples to be taken.

double sample_to_sample_delay (in)

Specifies the sample to sample interval delay, in seconds.

double timeout (in)

Specifies the trigger arm timeout, in second.

int ignore_trigger_count (in)

Specifies the number of triggers to be ignored, before performing measurement.
If ignore_trigger_count = 0, first trigger is a valid trigger.
If ignore_trigger_count = 2, module will ignore the first two triggers and only begin measuring after third trigger.

Descriptions

This function arms the specified SMU pin to prepare for voltage measurement in an array upon triggered.

Core Library Function Mapping

1. AemDCPwr_ConfigureInputTriggerSelect
2. AemDCPwr_ConfigureTriggerEdgeLevelExtra
3. AemDCPwr_ConfigureMeasureMode (mode = 3)

TrigArm_ReadVoltageArray

Synopsis

long TrigArm_ReadVoltageArray (pinAlias, trigSource, trigMode, meas_delay_after_trig, sampleSize, sample_to_sample_delay, ignore_trigger_count)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

int trigSource (in)

Specifies the triggering source.

int trigMode (in)

Specifies the triggering mode.

double meas_delay_after_trig (in)

Specifies the measure delay after being triggered, in seconds.

int sampleSize (in)

Specifies the number of samples to be taken.

double sample_to_sample_delay (in)

By default, upon power-up interval_s is 1us. Resolution of interval_s is 1us. Unit of interval_s is in seconds and therefore automatically rounded to the closest us. Minimum interval_s is 1us and maximum interval_s is 1ms.

int ignore_trigger_count (in)

Specifies the number of triggers to be ignored, before performing measurement.
If ignore_trigger_count = 0, first trigger is a valid trigger.
If ignore_trigger_count = 2, module will ignore the first two triggers and only begin measuring after third trigger.

Descriptions

This function arms the specified SMU pin to prepare for voltage measurement in an array upon triggered. This function calculates the NPLC according to the sampleSize and sample_to_sample_delay before arms the SMU pin.

Core Library Function Mapping

1. AemDCPwr_ConfigureSamplingTime
2. AemDCPwr_ConfigureAcquireArrayInterval
3. AemDCPwr_ConfigureInputTriggerSelect
4. AemDCPwr_ConfigureTriggerEdgeLevel(for module AM430e) or AemDCPwr_ConfigureTriggerEdgeLevelExtra for other module
5. AemDCPwr_ConfigureMeasureMode (mode = 3)

TrigGet_ReadVoltageArray

Synopsis

long TrigGet_ReadVoltageArray (pinAlias, sampleSize, rawSamples, min, max, average)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

int * sampleSize (out)

Returns the number of samples captured.

double * rawSamples (out)

Returns an array of sampled current.

double* min (out)

Returns the minimum value of the array.

double* max (out)

Returns the maximum value of the array.

double* average (out)

Returns the average value of the array.

Descriptions

This function retrieves the measured voltage when measurement is completed.

Core Library Function Mapping

1. AemDCPwr_AcquireVoltageArray

TrigArm_Release

Synopsis

long TrigArm_Release (pinAlias)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

Descriptions

This function clears the trigger status.

Core Library Function Mapping

1. AemDCPwr_ClearAcquireRecordLength
2. AemDCPwr_ConfigureInputTriggerSelect (triggerInput = 0)

ConfigureTriggerEdgeLevel

Synopsis

long ConfigureTriggerEdgeLevel (moduleAlias, trigSource, trigMode)

Arguments

String ^ moduleAlias (in)

Specifies the alias of the selected module.

int trigSource (in)

Specifies the triggering source.

trigSource	Description
0	Tri-state the trigger source
1	PXI backplane trigger line 0
2	PXI backplane trigger line 1
3	PXI backplane trigger line 2
4	PXI backplane trigger line 3
5	PXI backplane trigger line 4
6	PXI backplane trigger line 5
7	PXI backplane trigger line 6
8	PXI backplane trigger line 7
9	PXI backplane PXI_LBL6 signal
10	PXI backplane PXI_LBR6 signal
11	PXI backplane star trigger line
12	PXI backplane differential start trigger PXIE_DSTARA
13	PXI backplane differential start trigger PXIE_DSTARB
14	PXI backplane differential start trigger PXIE_DSTARC
19	Software trigger signal 0
20	Software trigger signal 1
21	Software trigger signal 2
22	Software trigger signal 3
23	External trigger port

Table 17: trigSource

int trigMode (in)

Specifies the triggering mode.

trigMode	Description
0	Channel will be triggered when rising edge is detected. This is the default mode
1	Channel will be triggered when falling edge is detected
2	Channel will be triggered when trigger signal is below a TTL logic level
3	Channel will be triggered when trigger signal exceeds logic level high

Table 18: trigMode

Core Library Function Mapping

1. AemDCPwr_ConfigureTriggerEdgeLevelExtra

ConfigureTriggerEdgeLevelExtra

Synopsis

long ConfigureTriggerEdgeLevelExtra (moduleAlias, trigSource, trigMode, ignore_trigger_count)

Arguments

String ^ moduleAlias (in)

Specifies the alias of the selected module.

int trigSource (in)

Specifies the triggering source.

trigSource	Description
0	Tri-state the trigger source
1	PXI backplane trigger line 0
2	PXI backplane trigger line 1
3	PXI backplane trigger line 2
4	PXI backplane trigger line 3
5	PXI backplane trigger line 4
6	PXI backplane trigger line 5
7	PXI backplane trigger line 6
8	PXI backplane trigger line 7
9	PXI backplane PXI_LBL6 signal
10	PXI backplane PXI_LBR6 signal
11	PXI backplane star trigger line
12	PXI backplane differential start trigger PXIE_DSTARA
13	PXI backplane differential start trigger PXIE_DSTARB
14	PXI backplane differential start trigger PXIE_DSTARC
19	Software trigger signal 0
20	Software trigger signal 1
21	Software trigger signal 2
22	Software trigger signal 3
23	External trigger port

Table 19: trigSource

int trigMode (in)

Specifies the triggering mode.

trigMode	Description
0	Channel will be triggered when rising edge is detected. This is the default mode
1	Channel will be triggered when falling edge is detected
2	Channel will be triggered when trigger signal is below a TTL logic level
3	Channel will be triggered when trigger signal exceeds logic level high

Table 20: trigMode

int ignore_trigger_count (in)

Specifies the number of triggers to be ignored, before performing measurement.

Descriptions

This function clears the trigger status.

Core Library Function Mapping

1. AemDCPwr_ConfigureTriggerEdgeLevel

MapTriggerInToTriggerOut

Synopsis

long MapTriggerInToTriggerOut (moduleAlias, inputTerminal, outputTerminal)

Arguments

String ^ moduleAlias (in)

Specifies the alias of the selected module.

int inputTerminal (in)

Specifies the triggering line to be connected to outputTerminal.

int outputTerminal (in)

Specifies the triggering line to be connected to inputTerminal.

inputTerminal / outputTerminal	Description
0	Tri-state the trigger source
1	PXI backplane trigger line 0
2	PXI backplane trigger line 1
3	PXI backplane trigger line 2
4	PXI backplane trigger line 3
5	PXI backplane trigger line 4
6	PXI backplane trigger line 5
7	PXI backplane trigger line 6
8	PXI backplane trigger line 7
9	PXI backplane PXI_LBL6 signal
10	PXI backplane PXI_LBR6 signal
11	PXI backplane star trigger line
12	PXI backplane differential start trigger PXIE_DSTARA
13	PXI backplane differential start trigger PXIE_DSTARB
14	PXI backplane differential start trigger PXIE_DSTARC
19	Software trigger signal 0
20	Software trigger signal 1
21	Software trigger signal 2
22	Software trigger signal 3
23	External trigger port

Table 21: Input/ Output terminal selection

Core Library Function Mapping

1. AemDCPwr_MapTriggerInToTriggerOut

DriveSoftwareTrigger

Synopsis

long DriveSoftwareTrigger (moduleAlias, select, pulseWidth)

Arguments

String ^ moduleAlias (in)

Specifies the alias of the selected module.

int select (in)

Specifies the software trigger line.

select	Description
0	Software trigger signal 0
1	Software trigger signal 1
2	Software trigger signal 2
3	Software trigger signal 3

Table 22: Software Trigger Line Selection

double pulseWidth (in)

Specifies the pulse width of the trigger signal, in seconds.

Core Library Function Mapping

1. AemDCPwr_DriveSoftwareTrigger

ConfigureInputTriggerSelect

Synopsis

long ConfigureInputTriggerSelect (pinAlias, triggerOutput, delayS)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

int triggerOutput (in)

Specifies the trigger output signal.

triggerOutput	Description
0	Tri-state the trigger source
1	PXI backplane trigger line 0
2	PXI backplane trigger line 1
3	PXI backplane trigger line 2
4	PXI backplane trigger line 3
5	PXI backplane trigger line 4
6	PXI backplane trigger line 5
7	PXI backplane trigger line 6
8	PXI backplane trigger line 7
9	PXI backplane PXI_LBL6 signal
10	PXI backplane PXI_LBR6 signal
11	PXI backplane star trigger line
12	PXI backplane differential start trigger PXIE_DSTARA
13	PXI backplane differential start trigger PXIE_DSTARB
14	PXI backplane differential start trigger PXIE_DSTARC
19	Software trigger signal 0
20	Software trigger signal 1
21	Software trigger signal 2
22	Software trigger signal 3
23	External trigger port

Table 23 : Trigger Output

double delayS (in)

Specifies the delay inserted before generating the trigger output, after operation such as ConfigureSMUOutputTriggerMode, is executed.

Core Library Function Mapping

1. AemDCPwr_ConfigureInputTriggerSelect

ConfigureOutputTriggerSelect

Synopsis

long ConfigureOutputTriggerSelect (pinAlias, triggerOutput, delayS)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

int triggerOutput (in)

Specifies the trigger output signal.

triggerOutput	Description
0	Tri-state the trigger source
1	PXI backplane trigger line 0
2	PXI backplane trigger line 1
3	PXI backplane trigger line 2
4	PXI backplane trigger line 3
5	PXI backplane trigger line 4
6	PXI backplane trigger line 5
7	PXI backplane trigger line 6
8	PXI backplane trigger line 7
9	PXI backplane PXI_LBL6 signal
10	PXI backplane PXI_LBR6 signal
11	PXI backplane star trigger line
12	PXI backplane differential start trigger PXIE_DSTARA
13	PXI backplane differential start trigger PXIE_DSTARB
14	PXI backplane differential start trigger PXIE_DSTARC
19	Software trigger signal 0
20	Software trigger signal 1
21	Software trigger signal 2
22	Software trigger signal 3
23	External trigger port

Table 24 : Trigger Output

double delayS (in)

Specifies the delay inserted before generating the trigger output, after operation such as ConfigureSMUOutputTriggerMode, is executed.

Core Library Function Mapping

2. AemDCPwr_ConfigureOutputTriggerSelect

ConfigureSMUOutputTriggerMode

Synopsis

long ConfigureSMUOutputTriggerMode (pinAlias, mode)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

int mode (in)

Specifies the trigger mode.

mode	Description
0	No trigger action
1	Generate trigger output when source action is completed
2	Generate trigger output when measure action is completed
3	Generate trigger output when compliance hit
4	Generate trigger output when compliance hit exits
5	Generate trigger output when during source action

Table 25: Trigger Mode

Core Library Function Mapping

1. AemDCPwr_ConfigureSMUOutputTriggerMode

ConfigureSMUOutputTriggerPulseWidth

Synopsis

long ConfigureSMUOutputTriggerPulseWidth (pinAlias, pulseWidth)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

double pulseWidth (in)

Specifies the width of the trigger output pulse, in seconds.

Core Library Function Mapping

1. AemDCPwr_ConfigureSMUOutputTriggerPulseWidth

ConfigureSMUOutputTriggerDuringSource

Synopsis

long ConfigureSMUOutputTriggerPulseWidth (pinAlias, level, range, mode, edgeSetting)

Arguments

String ^ pinAlias (in)

Specifies the alias of the target SMU pin.

double level (in)

Specifies the threshold level of the channel where a trigger output is generated when this threshold level is hit, depending on edgeSetting.

double range (in)

Specifies the range of the threshold level.

int mode (in)

Specifies whether the threshold is voltage or current value.
Set to "0" for voltage.
Set to "1" for current.

int edgeSetting (in)

Specifies the condition on how to trigger output will be generated by the specified channel. For multi-channel operation, a trigger signal will be generated if any one of channels meets the trigger condition.

edgeSetting	Description
0	Channel will be triggered when rising edge is detected. This is the default mode
1	Channel will be triggered when falling edge is detected
2	Channel will be triggered when trigger signal is below a TTL logic level
3	Channel will be triggered when trigger signal exceeds logic level high

Table 26: Edge Settings

1. AemDCPwr_ConfigureSMUOutputTriggerPulseWidth

Section 6: Error Message

The table below shows the error message specific to wrapper library.

Error Code (Hex)	Description
0xAE100100	Load_DLL_File_Error
0xAE100101	Load_DLL_Function_Error
0xAE100102	Memory_Allocation_Error
0xAE100110	File_IO_Error
0xAE100013	Invalid_Setting
0xAE100020	Software_Timeout
0xAE100021	Hardware_Timeout
0xAE100022	Exceed_Max_Sample_Size
0xAE100024	No_TestHead
0xAE100025	No_TestSite
0xAE100026	API_Not_Supported
0xAE100210	Voltage_Out_Of_Range
0xAE100211	Current_Out_Of_Range
0xAE100212	Invalid_Resource_Pin
0xAE100213	ExtTriGet_Timeout
0xAE100214	ExtTriGet_Unarmed
0xAE100215	ExtTriGet_Untriggered

Table 27: Wrapper Library Error Code

Section 7: Revision History

1.0	FEB 2013	INITIAL RELEASE
1.1	JUNE 2013	ADDED NEW APIS
1.2	JULY 2013	REVISED CONFIGURESMUOUTPUTTRIGGERDURINGSOURCE REVISED DRIVESOFTWARETRIGGER ADDED CONFIGUREPULSE ADDED CONFIGUREEXTENDERMODULE ADDED STARTCONTINOUSPULSE ADDED STOPCONTINOUSPULSE ADDED MEASUREPULSEVI
1.3	OCT 2013	ADDED CONFIGUREOUTPUTENABLED
1.4	DEC 2013	ADDED EXTTRIGARM_READCURRENTARRAY, EXTTRIGARM_READVOLTAGEARRAY, TRIGARM_READCURRENTARRAY AND TRIGARM_READVOLTAGEARRAY OVERLOAD FUCNTIONS. ADDED CONFIGUREACQUIREARRAYINTERVAL REVIEW EXTTRIGARM FUNCTIONS, WRAP MAPTRIGGERINTOTRIGGEROUT
1.5	FEB 2014	ADDED NEW API CONFIGUREACQUIREARRAYBWLIMIT.

Section 8: Contact Us

To obtain service, warranty or technical assistance, please contact Aemulus.

Aemulus Corporation Sdn. Bhd.
Krystal Point, B-2-04, B-2-05, B-2-06 & B-2-07
303, Jalan Sultan Azlan Shah,
11900 Penang, Malaysia
Tel: +604 6446399
Fax: +604 6466799z

Web: www.aemulus.com
Email: enquiry@aemulus.com

Product specifications and descriptions in this document are subject to change without prior notice.