PCIe-GIE7x Series

Function Library Reference

Manual Rev.: 1.0

Revision Date: July 14, 2017

Part No: 50-11178-2000

Leading **EDGE COMPUTING**

Revision History

Revision	Release Date	Description of Change(s)
2.00	July 14, 2017	Initial Release

Preface

Disclaimer

The information in this document is subject to change without prior notice in order to improve reliability, design, and function and does not represent a commitment on the part of the manufacturer.

In no event will the manufacturer be liable for direct, indirect, special, incidental, or consequential damages arising out of the use or inability to use the product or documentation, even if advised of the possibility of such damages.

Environmental Responsibility

ADLINK is committed to fulfill its social responsibility to global environmental preservation through compliance with the European Union's Restriction of Hazardous Substances (RoHS) directive and Waste Electrical and Electronic Equipment (WEEE) directive. Environmental protection is a top priority for ADLINK. We have enforced measures to ensure that our products, manufacturing processes, components, and raw materials have as little impact on the environment as possible. When products are at their end of life, our customers are encouraged to dispose of them in accordance with the product disposal and/or recovery programs prescribed by their nation or company.

Trademarks

Product names mentioned herein are used for identification purposes only and may be trademarks and/or registered trademarks of their respective companies.

Conventions

Take note of the following conventions used throughout this manual to make sure that users perform certain tasks and instructions properly.

Additional information, aids, and tips that help users perform tasks.

Information to prevent ***minor*** physical injury, component damage, data loss, and/or program corruption when trying to complete a task.

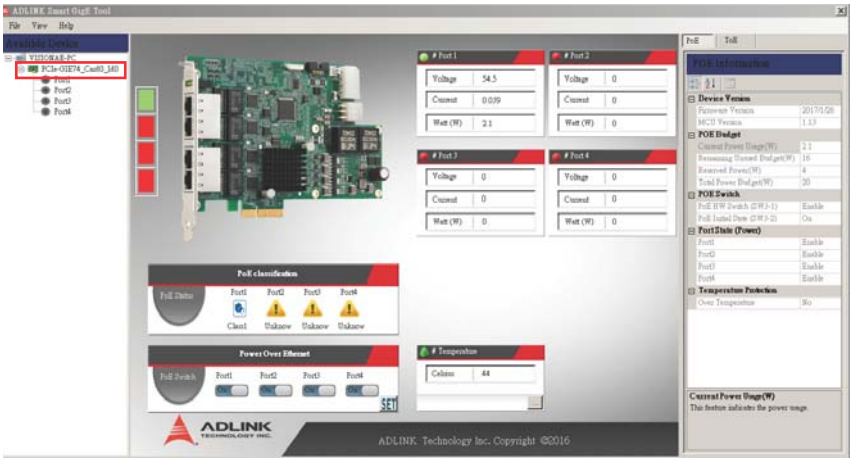
Information to prevent ***serious*** physical injury, component damage, data loss, and/or program corruption when trying to complete a specific task.

Table of Contents

This page intentionally left blank.

1 Introduction

The PCIe-GIE7X Function Library Reference provides necessary API details, where functions can develop power management, Trigger over Ethernet, and License Management applications in C/C++, C#/VB.Net. Once the driver is installed, the PCIe-GIE7X is listed in the ADLINK Smart GigE Utility, as shown.



1.1 Setting Up the Build Environment

All files are located in the directory \ADLINK\PCIe-GIE7X\Include

File Type	File	Description	Required by
Include	SmartPoE.h.	Header file required for all C/C++ applications	All applications using APIs
	SmartPoE.vb	Function definitions required for all VB.Net applications	
	SmartPoE.cs	Function definitions required for all C# applications	
DLL	SmartPoE.dll	Dynamic link library required for all applications	

File Type	File	Description	Required by
Library	SmartPoE.lib	Exports API function definitions, required for all Visual C/C++ applications	All C/C++ applications using APIs

2 Function Library

2.1 List of Functions

Category	Function	Cards
Device Control	SmartPoE_Register_Card	PCIe-GIE64, PCIe-GIE7X
	SmartPoE_Release_Card	
	SmartPoE_Get_ID	
	SmartPoE_Get_CPLDVersion	PCIe-GIE7X
	SmartPoE_Get_MCUVersion	
Power Management	SmartPoE_Power_Enable	PCIe-GIE64, PCIe-GIE7X
	SmartPoE_Get_Power_Enable	
	SmartPoE_Get_PoEstate	
	SmartPoE_Get_PoEPortLaststate	
	SmartPoE_Get_Temperature	
	SmartPoE_Get_HighTemperature	
	SmartPoE_Get_LowTemperature	
	SmartPoE_Set_HighTemperature	
	SmartPoE_Set_LowTemperature	
	SmartPoE_Get_PortStatus	
	SmartPoE_Get_PSEPortCurrent	
	SmartPoE_Get_PSEPortVoltage	
	SmartPoE_Get_PoEConsPowbudget	
	SmartPoE_Get_PoELeftPowbudget	
	SmartPoE_Get_PowerBudgetControl	
	SmartPoE_Set_PowerBudgetControl	
License Management	AD_InstallSecret	PCIe-GIE7X PRO
	AD_SetMasterSecret	
	AD_EncryptReadSegment	
	AD_EncryptComputeEnc	
	AD_EncryptAuthWritePageEnc	
	AD_ReadInstallSecretStatus	

Category	Function	Cards
Trigger over Ethernet	GIE_Set_ActionCommand	PCIe-GIE7X PRO
	GIE_Get_ActionCommand	
	GIE_Send_ActionCommand	
	GIE_Send_AllActionCommand	
	GIE_GetTriggerSource	
	GIE_SetTriggerSource	
	GIE_GetTriggerMode	
	GIE_SetTriggerMode	
	GIE_GetTriggerActivation	
	GIE_SetTriggerActivation	
	GIE_GetTriggerType	
	GIE_SetTriggerType	
	GIE_GetTriggerDebounce	
	GIE_SetTriggerDebounce	
	GIE_GetTriggerCount	
	GIE_ResetTriggerCount	

2.2 Function Library

2.2.1 Device Control Functions

SmartPoE_Register_Card

Initializes a specified device, should be called before other functions except those with no Number parameter.

Syntax

C/C++

```
short SmartPoE_Register_Card (unsigned short card_num)
```

Parameter(s)

Card_num:

Indicates the number of devices, beginning at 0 for the number of the first card, with the second card 1, and so on.

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-3    The count of calling SmartPoE_Register_Card
      need to be less than 32
-13   Driver is not found or there is no card
      exists
```

SmartPoE_Release_Card

Closes the device and releases all allocated resources, should be called before terminating the application.

Syntax

C/C++

```
short SmartPoE_Release_Card (unsigned short
wCardNumber)
```

Parameter(s)

wCardNumber:

Indicates the value returned from calling **SmartPoE_Register_Card**.

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-4    Call SmartPoE_Register_Card before other
      functions
```

SmartPoE_Get_ID

Initializes the hardware and software states of a PCI-bus data acquisition card, then returns a numeric card ID that corresponds to the initialized card. Register_Card must be called before any other PCIS-DASK library functions can be called for a particular card. The function initializes the card and variables internal to the PCIS-DASK library. Because PCI-bus data acquisition cards meet plug-and-play specifi-

cations, the base address (pass-through address) and IRQ level are assigned directly by the system BIOS

Syntax

C/C++

```
short SmartPoE_Get_ID (unsigned short wCard-
Number, unsigned short *wID)
```

Parameter(s)

wCardNumber:

Acquired from the return value of SmartPoE_Register_Card

wID:

Indicates card ID set by S1 switch, a value from 0 to 15. Card ID and S1 switch settings correlate as shown

Card ID	S1 switch			
	4	3	2	1
0	off	off	off	off
1	off	off	off	on
2	off	off	on	off
3	off	off	on	on
4	off	on	off	off
5	off	on	off	on
6	off	on	on	off
7	off	on	on	on
8	on	off	off	off
9	on	off	off	on
10	on	off	on	off
11	on	off	on	on
12	on	on	off	off
13	on	on	off	on
14	on	on	on	off
15	on	on	on	on

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card
```

SmartPoE_Get_CPLDVersion

Retrieves CPLD version.

Syntax

C/C++

```
short SmartPoE_Get_CPLDVersion(unsigned short
wCardNumber,    unsigned    long    *wVersion1,
unsigned long *wVersion2 )
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card

wVersion1:

7:0 Date

15:8 Month

31:16 Year

wVersion2:

7:0 Minute

15:8 Hour

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card
```

SmartPoE_Get_MCUVersion

Retrieves the PoE MCU version.

Syntax

C/C++

```
short SmartPoE_Get_MCUVersion (unsigned short
wCardNumber, unsigned short wVersion)
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

wVersion:

Bit 0 to 3: minor number

Bit 4 to 7: major number

For example V1.3

Bit 0 to 3: minor number, in this case 3

Bit 4 to 7: major number, in this case 1

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card
```

2.2.2 Power Management Functions

SmartPoE_Power_Enable

Controls power auto/off.

Syntax

C/C++

```
short SmartPoE_Power_Enable (unsigned short
wCardNumber, unsigned short wEnPort1, unsigned
```

```
short wEnPort2, unsigned short wEnPort3,
unsigned short wEnPort4)
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

wEnPort1, wEnPort2, wEnPort3, wEnPort4:

Indicates power mode of the Smart PoE port with:

0: power off

1: auto mode (enables port line detection)

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2   An invalid wCardNumber parameter,
     wCardNumber is less than 32
-5   The function is not supported by this card
```

SmartPoE_Get_Power_Enable

Retrieves power auto/off.

Syntax

C/C++

```
short SmartPoE_Get_Power_Enable (unsigned
short wCardNumber, unsigned short *wEnPort1,
unsigned short *wEnPort2, unsigned short
*wEnPort3, unsigned short *wEnPort4)
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

wEnPort1, wEnPort2, wEnPort3, wEnPort4:

Indicates power mode of the Smart PoE port with:

0: power off

1: auto mode (enables port line detection)

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card
```

SmartPoE_Get_PoEstate

Retrieves PoE state.

Syntax

C/C++

```
short SmartPoE_Get_PoEstate (unsigned short
wCardNumber, unsigned char *bPoEstatePA0,
unsigned char *bPoEstatePA1, unsigned char
*bPoEstatePowerBudget, unsigned char
*bPoEstatePA4, unsigned char *bPoEstatePA3,
unsigned char *bPoEstatePF0)
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

wPoEstate:

bPoEstatePA0: (PA0)

0: PoE-PSE HW setting disabled

1: PoE-PSE HW setting enabled

bPoEstatePA1: (PA1)

0: PoE-PSE should be turned off when powered on

1: PoE-PSE should be turned on when powered on.

bPoEstatePowerBudget: (PA5 and PB1)

00: power budget = 61.6W

01: power budget = 120W

10: power budget = 20W

11: power budget = 61.6W

bPoEstatePA4: (PA4)

0: 52V off, PoE-PSE is down

1: 52V on, PoE-PSE is on

bPoEstatePA3: (PA3) INT state of thermal sensor (Low Active)

0: over temperature interrupt arise

1: no interrupt

bPoEstatePF0: (PF0) Port numbers

0: 4 PoE ports

1: 2 PoE ports (port 1 and port 2)

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card
      SmartPoE_Get_MCUVersion
```

SmartPoE_Get_PoEPortLaststate

Retrieves the PoEPort Last State. MCU recovers POE-PSE Port Last State when PSE is off/on.

Syntax

C/C++

```
Short SmartPoE_Get_POEPortLaststate (unsigned
short wCardNumber, unsigned short * wPOEstate)
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

wPOEstate:

Bit 0:

0 : port 1 power not enabled

1 : port 1 power enabled

Bit 1:

0 : port 2 power not enabled

1 : port 2 power enabled

Bit 2: (if there are only 2 POE ports, this bit is always 0)

0 : port 3 power not enabled

1 : port 3 power enabled

Bit 3: (if there are only 2 POE ports, this bit is always 0)

0 : port 4 power not enabled, or there are only 2 POE ports

1 : port 4 power enabled

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card
      SmartPoE_Get_MCUVersion
```

SmartPoE_Get_Temperature

Retrieves thermal temperature.

Syntax

C/C++

```
short SmartPoE_Get_Temperature (unsigned short
wCardNumber, double *wTemperature)
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

wTemperature:

Thermal temperature in °C.

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card
      SmartPoE_Get_MCUVersion
```

SmartPoE_Get_HighTemperature

Retrieves the High Temperature setting.

Syntax

C/C++

```
short SmartPoE_Get_HighTemperature (unsigned
short wCardNumber, unsigned short *wTempera-
ture)
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

wHighTemperature:

High Temperature setting, not exceeding 150°C.

1 byte data in 1°C unit.

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card
      SmartPoE_Get_MCUVersion
```

SmartPoE_Get_LowTemperature

Retrieves Low Temperature setting (cannot exceed High Temperature setting).

Syntax

C/C++

```
short SmartPoE_Get_LowTemperature (unsigned short wCardNumber, unsigned short * wLowTemperature)
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

wLowTemperature:

Low Temperature setting (cannot exceed High Temperature setting

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2   An invalid wCardNumber parameter,
     wCardNumber is less than 32
-5   The function is not supported by this card
     SmartPoE_Get_MCUVersion
```

SmartPoE_Set_HighTemperature

High Temperature setting.

Syntax

C/C++

```
short SmartPoE_Set_HighTemperature (unsigned short wCardNumber, unsigned short wHighTemperature)
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

wHighTemperature:

High Temperature setting, more than or equal to 80°C, not exceeding 130°C, and at least 10°C higher than the Low Temperature setting.

1 byte data in 1°C unit.

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card
      SmartPoE_Get_MCUVersion
-11   The input argument(s) exceed the allowed
      range.
```

SmartPoE_Set_LowTemperature

Low Temperature setting.

Syntax

C/C++

```
short SmartPoE_Set_LowTemperature (unsigned
short wCardNumber, unsigned short wLowTempera-
ture)
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

wLowTemperature:

Low Temperature setting, equal to or exceeding 70°C and at least 10°C lower than the High Temperature setting.

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card
      SmartPoE_Get_MCUVersion
-11   The input argument(s) exceed the allowed
      range.
```


NOTE:

High Temperature and Low Temperature settings must be more than 10°C apart.

SmartPoE_Get_PortStatus

Retrieves Port Status.

Syntax

C/C++

```
short SmartPoE_Get_PortStatus (unsigned short
wCardNumber, unsigned short PortNumber,
unsigned char * bstateClass, unsigned char *
bstatePowerGood)
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

PortNumber:

Indicates port number from 1 to 4.

wstate:

bstateClass:

0:Unknow

1:Class 1

2:Class 2

3:Class 3

4:Class 4

5:Class 5

6:Class 0

7:Current limit

bstateClass	PoE CLASS
000	Unknown
001	1
010	2
011	3
100	4
101	5
110	0
111	Current limit

bstatePowerGood:

1: PowerGood

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card
      SmartPoE_Get_MCUVersion
```

SmartPoE_Get_PSEPortCurrent

Retrieves port current.

Syntax

C/C++

```
short     SmartPoE_Get_PSEPortCurrent(unsigned
short wCardNumber, unsigned short PortNumber
double *wCurrent)
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

PortNumber:

Indicates port number from 1 to 4.

wCurrent:

Port current

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card
      SmartPoE_Get_MCUVersion
```

SmartPoE_Get_PSEPortVoltage

Retrieves port voltage.

Syntax

C/C++

```
short   SmartPoE_Get_PSEPortVoltage   (unsigned
short wCardNumber, unsigned short PortNumber
double *wVoltage)
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

PortNumber:

Indicates port number from 1 to 4.

wVoltage:

Port voltage

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card
      SmartPoE_Get_MCUVersion
```

SmartPoE_Get_PoEConsPowbudget

Retrieves the Consumed Power budget.

Syntax

C/C++

```
short    SmartPoE_Get_POEConsPowbudget(unsigned
short wCardNumber, double * wPower)
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

wPower:

Consumed Power budget in W.

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card
      SmartPoE_Get_MCUVersion
```

SmartPoE_Get_PoELeftPowbudget

Retrieves the Left Power budget.

Syntax

C/C++

```
short   SmartPoE_Get_POELeftPowbudget(unsigned
short wCardNumber, double * wPower)
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

wPower:

Left Power budget in W.

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card
      SmartPoE_Get_MCUVersion
```

SmartPoE_Get_PowerBudgetControl

Retrieves power budget control status.

Syntax

C/C++

```
short SmartPoE_Get_PowerBudgetControl(unsigned
short wCardNumber, unsigned short * wMode)
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

wMode:

Mode: 0 (Turn off power budget control)

Mode: 1 (Turn on power budget control)

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card
      SmartPoE_Get_MCUVersion
```

SmartPoE_Set_PowerBudgetControl

Sets power budget control status.

Syntax

C/C++

```
short SmartPoE_Set_PowerBudgetControl(unsigned
short wCardNumber, unsigned short  wMode)
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

wMode:

Mode: 0 (Turn off power budget control.)

Mode: 1 (Turn on power budget control.)

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card
      SmartPoE_Get_MCUVersion
```

2.2.3 License Management Functions

AD_InstallSecret

Installs secret on DS28C22, and can, optionally, lock the secret (irreversible).

Syntax

C/C++

```
short AD_InstallSecret(unsigned short wCard-
Number, unsigned char *MasterSecret, bool
lockbit)
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

MasterSecret:

32 byte buffer with new secret (or partial).

LockBit:

Allows secret to be write-protected, where it is read as protected by hardware design. Once set, secret protection cannot be reset.

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card
      SmartPoE_Get_MCUVersion
-12    An invalid secret. (SetMastSecret error)
-15    Secret has been locked.
```

AD_SetMasterSecret

Sets secret on DS28C22.

Syntax

C/C++

```
short AD_SetMasterSecret(unsigned short wCard-
Number, unsigned char *MasterSecret)
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

MasterSecret:

32 byte buffer with new secret (or partial).

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card
      SmartPoE_Get_MCUVersion
-12    An invalid secret. (SetMastSecret error)
```

AD_EncryptReadSegment

Performs an encrypted read DS28C22 on an encrypted segment of memory, starting on the specified page at the specified segment offset.

Syntax

C/C++

```
short AD_EncryptReadSegment(unsigned short
wCardNumber, unsigned char *data, unsigned
char *romid, unsigned char *manid, unsigned
char *read_mac, unsigned char *read_challenge)
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

data:

Buffer for data to read (32-byte)

romid:

Buffer containing device 8-byte ROM ID used in encryption.

manid:

Buffer for 2-byte manufacturer ID used in encryption calculation

read_mac:

Buffer containing 32 bytes of read_mac

read_challenge:

Buffer containing 8 bytes of challenge

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card
      SmartPoE_Get_MCUVersion
-12    An invalid secret. (SetMastSecret error)
```

AD_EncryptComputeEnc

Computes encode data.

Syntax

C/C++

```
short AD_EncryptComputeEnc_ex2(unsigned short
wCardNumber, unsigned char *new_data, unsigned
char *romid, unsigned char *manid, unsigned
char *read_mac, unsigned char *challenge,
unsigned char *enc_data, unsigned char *
chk_mac)
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

numBytesTot:

Total bytes to be written to page.

enc_data:

Buffer containing 32 bytes of data to be written

old_data:

Buffer containing 32 bytes of data currently in that segment (required for authentication calc.)

romid:

Buffer containing device 8-byte ROM ID used in encryption.

manid:

Buffer for 2-byte manufacturer ID used in encryption calculation

check_mac:

Buffer containing 32 bytes of check_mac

read_challenge:

Buffer containing 8 bytes of challenge

lockdata:

Once set, data protection mode cannot be reset.

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2   An invalid wCardNumber parameter,
     wCardNumber is less than 32
-5   The function is not supported by this card
     SmartPoE_Get_MCUVersion
-11   An error in lock function.
-70   An error in encryption authentication.
```

AD_EncryptAuthWritePageEnc

Writes up to 32 bytes in WHOLE block segment(s) to EEPROM on provided page using authenticated encryption.

Syntax

C/C++

```
short AD_EncryptAuthWritePageEnc(unsigned short wCardNumber, unsigned short numBytesTot, unsigned char *enc_data, unsigned char *old_data, unsigned char *romid, unsigned char *manid, unsigned char *chk_mac, unsigned char *challenge, bool lockdata)
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

numBytesTot:

Total bytes to be written to page.

enc_data:

Buffer containing 32 bytes of data to be written

old_data:

Buffer containing 32 bytes of data currently in that segment (required for authentication calc.)

romid:

Buffer containing device 8-byte ROM ID used in encryption.

manid:

Buffer for 2-byte manufacturer ID used in encryption calculation

check_mac:

Buffer containing 32 bytes of check_mac

read_challenge:

Buffer containing 8 bytes of challenge

lockdata:

Once set, data protection mode cannot be reset.

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card
      SmartPoE_Get_MCUVersion
-11    An error in lock function.
-70    An error in encryption authentication.
```

AD_ReadInstallSecretStatus

Retrieves how each individual user memory page can be protected.

Syntax

C/C++

```
short      AD_ReadInstallSecretStatus(unsigned
short  wCardNumber,  byte*  InstallBit,  byte*
LockDataBit)
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

InstallBit:

0: Non-burned card

1: Burned card

2: Burned card and data locked

LockDataBit:

0: Secret non-locked

1: Secret locked

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card
      SmartPoE_Get_MCUVersion
```

2.2.4 Trigger over Ethernet Functions

GIE_Set_ActionCommand

Sets an ACTION command over the selected port.

Syntax

C/C++

```
short  GIE_Set_ActionCommand(U16  wCardNumber,
U16 PortNumber, U32 gActionDeviceKey, U32 gAc-
tionGroupKey, U32 gActionGroupMask);
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

PortNumber:

Indicates port number from 1 to 4.

gActionDeviceKey:

Device Key of the command.

gActionGroupKey:

Group Key of the command

gActionGroupMask:

Group Mask of the command

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card.
-77    An invalid IP address.
```

GIE_Get_ActionCommand

Gets an ACTION command over the selected port.

Syntax

C/C++

```
short  GIE_Get_ActionCommand(U16  wCardNumber,
U16  PortNumber,  U32  *gActionDeviceKey,  U32
*gActionGroupKey, U32 *gActionGroupMask);
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

PortNumber:

Indicates port number from 1 to 4.

gActionDeviceKey:

Device Key of the command.

gActionGroupKey:

Group Key of the command

gActionGroupMask:

Group Mask of the command

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card.
```

GIE_Send_ActionCommand

Sends a software ACTION command over the selected port.

Syntax

C/C++

```
short       GIE_Send_SoftwareActionCommand(U16
wCardNumber, U16 PortNumber);
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

PortNumber:

Indicates port number from 1 to 4.

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card.
```

GIE_Send_AllActionCommand

Sends a software ACTION command over all ports.

Syntax

C/C++

```
short    GIE_Send_AllSoftwareActionCommand(U16
wCardNumber);
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card.
```

GIE_GetTriggerSource

Acquires Trigger over Ethernet function source.

Syntax

C/C++

```
short   GIE_GetTriggerSource(U16   wCardNumber,
U16 *Source);;
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

State:

0: Software trigger source

1: External hardware trigger source

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card.
```

GIE_SetTriggerSource

Sets the Trigger over Ethernet function source.

Syntax

C/C++

```
short   GIE_SetTriggerSource(U16   wCardNumber,
U16 Source);;
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

State:

0: Software trigger source

1: External hardware trigger source

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card.
-11) An invalid state parameter.
```

GIE_GetTriggerMode

Acquires Trigger over Ethernet function status.

Syntax

C/C++

```
short GIE_GetTriggerMode(U16 wCardNumber, U16
*Mode);;
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

Mode:

0: Software trigger source

1: External hardware trigger source

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2   An invalid wCardNumber parameter,
     wCardNumber is less than 32
-5   The function is not supported by this card.
-11) An invalid state parameter.
```

GIE_SetTriggerMode

Acquires Trigger over Ethernet function status.

Syntax

C/C++

```
short GIE_GetTriggerMode(U16 wCardNumber, U16
*Mode);;
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

Mode:

0: Software trigger source

1: External hardware trigger source

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2   An invalid wCardNumber parameter,
     wCardNumber is less than 32
-5   The function is not supported by this card.
-11) An invalid state parameter.
```

GIE_GetTriggerActivation

Acquires Trigger activation mode of ToE, specifying that the source trigger is considered valid on the rising edge or falling edge.

Syntax

C/C++

```
short GIE_GetTriggerActivation (U16 wCardNumber, U16 *Activation);;
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

Activation:

1: Rising edge

2: Falling edge

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card.
```

GIE_SetTriggerActivation

Sets Trigger activation mode for ToE, specifying that the source trigger is considered valid on the rising edge or falling edge.

Syntax

C/C++

```
short GIE_SetTriggerActivation (U16 wCardNumber, U16 Activation);;
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

Activation:

1: Rising edge

2: Falling edge

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card.
-11    This is an invalid state parameter.
```

GIE_GetTriggerType

Acquires Trigger type, where 4 to 4 mode indicates that each pin of the DI performs corresponding port action command to active, and 1 to 4 mode indicates that DI_0 performs all port action commands to active.

Syntax

C/C++

```
short GIE_GetTriggerType(U16 wCardNumber, U16
*Type);;
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

Type:

0: 4 to 4 mode

1: 1 to 4 mode

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card.
```

GIE_SetTriggerType

Sets Trigger type, where 4 to 4 mode indicates that each pin of the DI performs corresponding port action command to active, and 1 to 4 mode indicates that DI_0 performs all port action commands to active.

Syntax

C/C++

```
short GIE_SetTriggerType(U16 wCardNumber, U16
Type);;
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

Type:

0: 4 to 4 mode

1: 1 to 4 mode

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card.
-11) An invalid state parameter.
```

GIE_GetTriggerDebounce

Acquires Trigger debounce time for filtering the external trigger.

Syntax

C/C++

```
short GIE_GetTriggerDebounce (U16 wCardNumber,
U32 * Debounce);;
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

Debounce:

1µs to 200000µs

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card.
```

GIE_SetTriggerDebounce

Sets trigger denounce time for filtering the external trigger.


NOTE:

When the board connects with different trigger source types (voltage and frequency), photocouplers onboard experience slightly different rising and falling times, and, thus, the trigger debounce time is 25 to 35 ns longer than the time period (reciprocal of frequency) of the trigger source.

Syntax

C/C++

```
short GIE_SetTriggerDebounce (U16 wCardNumber,
U32 Debounce);;
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

Debounce:

1µs to 200000µs

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2   An invalid wCardNumber parameter,
     wCardNumber is less than 32
-5   The function is not supported by this card.
-11) An invalid state parameter.
```

GIE_GetTriggerCount

Acquires Trigger count, directing 16-bit counter to count triggers originating from hardware or software and ToE commands sent from the PCIe-GIE7x PRO.

Syntax

C/C++

```
short GIE_GetTriggerCount(U16 wCardNumber, U16
PortNumber, U16 *TriggerCount, U16 *Trig-
gerSentCount);
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

PortNumber:

Indicates port number from 1 to 4.

TriggerCount:

Number of triggers from hardware trigger or software trigger (Valid value: 0 to 65535)

TriggerSentCount:

Number of action commands from PCIe-GIE7x PRO (Valid value: 0 to 65535)

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card.
```

GIE_ResetTriggerCount

Resets Trigger count, with trigger counters reset to 0.

Syntax

C/C++

```
short GIE_ResetTriggerCount(U16 wCardNumber);
```

Parameter(s)

wCardNumber:

Retrieved from the return value of SmartPoE_Register_Card.

Return Code

No error occurs if return value >= 0, and if negative value, as follows

```
-2    An invalid wCardNumber parameter,
      wCardNumber is less than 32
-5    The function is not supported by this card.
```

Important Safety Instructions

For user safety, please read and follow all instructions, Warnings, Cautions, and Notes marked in this manual and on the associated device before handling/operating the device, to avoid injury or damage.

S'il vous plaît prêter attention stricte à tous les avertissements et mises en garde figurant sur l'appareil , pour éviter des blessures ou des dommages.

- Read these safety instructions carefully
- Keep the User's Manual for future reference
- Read the Specifications section of this manual for detailed information on the recommended operating environment
- The device can be operated at an ambient temperature of 55ºC (with DC supply) and 50ºC (with adapter supply);
- When installing/mounting or uninstalling/removing device; or when removal of a chassis cover is required for user servicing (See "Function Library" on page 3.):
  - Turn off power and unplug any power cords/cables
  - Reinstall all chassis covers before restoring power
- To avoid electrical shock and/or damage to device:
  - Keep device away from water or liquid sources
  - Keep device away from high heat or humidity
  - Keep device properly ventilated (do not block or cover ventilation openings)
  - Always use recommended voltage and power source settings
  - Always install and operate device near an easily accessible electrical outlet
  - Secure the power cord (do not place any object on/over the power cord)
  - Only install/attach and operate device on stable surfaces and/or recommended mountings
- If the device will not be used for long periods of time, turn off and unplug from its power source

- Never attempt to repair the device, which should only be serviced by qualified technical personnel using suitable tools
- A Lithium-type battery may be provided for uninterrupted backup or emergency power.

Risk of explosion if battery is replaced with one of an incorrect type; please dispose of used batteries appropriately.
Risque d'explosion si la pile est remplacée par une autre de type incorrect. Veuillez jeter les piles usagées de façon appropriée.

- The device must be serviced by authorized technicians when:
  - The power cord or plug is damaged
  - Liquid has entered the device interior
  - The device has been exposed to high humidity and/or moisture
  - The device is not functioning or does not function according to the User's Manual
  - The device has been dropped and/or damaged and/or shows obvious signs of breakage
- Disconnect the power supply cord before loosening the thumbscrews and always fasten the thumbscrews with a screwdriver before starting the system up
- It is recommended that the device be installed only in a server room or computer room where access is:
  - Restricted to qualified service personnel or users familiar with restrictions applied to the location, reasons therefor, and any precautions required
  - Only afforded by the use of a tool or lock and key, or other means of security, and controlled by the authority responsible for the location
- If PoE (Power over Ethernet) is enabled for the device, the system can ONLY be deployed indoors. Unless otherwise noted, the PoE system is NOT designed to withstand the rigors of outdoor use.

	**BURN HAZARD** Touching this surface could result in bodily injury. To reduce risk, allow the surface to cool before touching. ***RISQUE DE BRÛLURES*** *Ne touchez pas cette surface, cela pourrait entraîner des blessures.* *Pour éviter tout danger, laissez la surface refroidir avant de la toucher.*

This page intentionally left blank.

Getting Service

Ask an Expert: http://askanexpert.adlinktech.com

ADLINK Technology, Inc.
9F, No.166 Jian Yi Road, Zhonghe District
New Taipei City 235, Taiwan
Tel: +886-2-8226-5877
Fax: +886-2-8226-5717
Email: service@adlinktech.com

Ampro ADLINK Technology, Inc.
5215 Hellyer Avenue, #110
San Jose, CA 95138, USA
Tel: +1-408-360-0200
Toll Free: +1-800-966-5200 (USA only)
Fax: +1-408-360-0222
Email: info@adlinktech.com

ADLINK Technology (China) Co., Ltd.
300 Fang Chun Rd., Zhangjiang Hi-Tech Park
Pudong New Area, Shanghai, 201203 China
Tel: +86-21-5132-8988
Fax: +86-21-5132-3588
Email: market@adlinktech.com

LiPPERT ADLINK Technology GmbH
Hans-Thoma-Strasse 11
D-68163 Mannheim, Germany
Tel: +49-621-43214-0
Fax: +49-621 43214-30
Email: emea@adlinktech.com

Please visit the Contact page at **www.adlinktech.com** for information on how to contact the ADLINK regional office nearest you: