

Ametek_LIB

Generated by Doxygen 1.8.20

1 File Index	1
1.1 File List	1
2 File Documentation	3
2.1 C:/Users/jai_prajapati/Documents/SourceLibraries/Serial_LIB/Ametek_LIB/AMETEK_LIB.c File Reference	3
2.1.1 Detailed Description	4
2.1.2 Function Documentation	5
2.1.2.1 GetCurr()	5
2.1.2.2 GetStatus_ERRs()	5
2.1.2.3 GetStatus_ESR()	5
2.1.2.4 GetStatus_OUT()	5
2.1.2.5 GetStatus_PROT()	6
2.1.2.6 GetStatus_SCPI()	6
2.1.2.7 GetStatus_TRIP()	6
2.1.2.8 GetVoltage()	6
2.1.2.9 SelfTest()	7
2.1.2.10 SetDelay()	7
2.1.2.11 SetFold()	7
2.1.2.12 SetIsolation()	7
2.1.2.13 SetSense()	8
2.1.2.14 SetState()	8
2.2 C:/Users/jai_prajapati/Documents/SourceLibraries/Serial_LIB/Ametek_LIB/AMETEK_LIB.h File Reference	8
2.2.1 Detailed Description	9
2.2.2 Function Documentation	10
2.2.2.1 GetCurr()	10
2.2.2.2 GetStatus_ERRs()	10
2.2.2.3 GetStatus_ESR()	10
2.2.2.4 GetStatus_OUT()	10
2.2.2.5 GetStatus_PROT()	11
2.2.2.6 GetStatus_SCPI()	11
2.2.2.7 GetStatus_TRIP()	11
2.2.2.8 GetVoltage()	11
2.2.2.9 SelfTest()	12
2.2.2.10 SetDelay()	12
2.2.2.11 SetFold()	12
2.2.2.12 SetIsolation()	12
2.2.2.13 SetSense()	13
2.2.2.14 SetState()	13
Index	15

Chapter 1

File Index

1.1 File List

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

C:/Users/jai_prajapati/Documents/SourceLibraries/Serial_LIB/Ametek_LIB/AMATEK_LIB.h	??
C:/Users/jai_prajapati/Documents/SourceLibraries/Serial_LIB/Ametek_LIB/AMETEK_LIB.c	
Serial communication wrapper for Ametek used in RCM8 Run-In-Tester	3
C:/Users/jai_prajapati/Documents/SourceLibraries/Serial_LIB/Ametek_LIB/AMETEK_LIB.h	8

Chapter 2

File Documentation

2.1 C:/Users/jai_prajapati/Documents/SourceLibraries/Serial_LIB/↵ Ametek_LIB/AMETEK_LIB.c File Reference

Serial communication wrapper for Ametek used in RCM8 Run-In-Tester.

```
#include "toolbox.h"  
#include <userint.h>  
#include <ansi_c.h>  
#include "AMETEK_LIB.h"
```

Functions

- int [Initialize_AMETEK_LIB](#) (char errmsg[ERRLEN])
Initialize Ametek library. Requires SerialComm_LIB to be previously initialized and configured.
- int [GetStatus_ESR](#) (char errmsg[ERRLEN])
Get Event status register.
- int [GetStatus_SCPI](#) (char errmsg[ERRLEN])
Get SCPI status.
- int [GetStatus_PROT](#) (char errmsg[ERRLEN])
Get protection fault status.
- int [GetStatus_ERRs](#) (char errmsg[ERRLEN])
Gets the error status.
- int [GetStatus_OUT](#) (char errmsg[ERRLEN])
Get protection falut status.
- double [GetStatus_TRIP](#) (char errmsg[ERRLEN])
Get tripped status.
- double [GetVoltage](#) (char errmsg[ERRLEN])
Get voltage output level.
- double [GetCurr](#) (char errmsg[ERRLEN])
Get current output level.
- int [SetVolt](#) (double Volts, char errmsg[ERRLEN])
Sets the voltage on the PSU from the paramter given.

- int **SetLimit_Curr** (double Current, char errmsg[ERRLEN])
Sets the current limit on the PSU from the paramter given.
- int **SetFold** (int Type, char errmsg[ERRLEN])
Sets the protection type on the PSU from the paramter given.
- int **SetPolarity** (int Pol, char errmsg[ERRLEN])
Sets the polarity of the PSU from the paramter given.
- int **SetSense** (int Sense, char errmsg[ERRLEN])
Sets the sense relay signal open/closed on the PSU from the paramter given.
- int **SetState** (int State, char errmsg[ERRLEN])
Sets the output on the PSU from the paramter given.
- int **SetIsolation** (int Iso, char errmsg[ERRLEN])
Sets the isolation relay control signal on the PSU from the parameter given.
- int **SetDelay** (double Time, char errmsg[ERRLEN])
Sets the delay for fault reporting on the PSU from the paramter given.
- int **InitPSU** (double Volts, double Curr, char errmsg[ERRLEN])
Intializes the PSU based off the inputs.
- int **SelfTest** (char errmsg[ERRLEN])
Let the PSU Self-Check.
- int **ClearPSUStatus** (char errmsg[ERRLEN])
Clears PSU registers.
- int **ResetPSU** (char errmsg[ERRLEN])
Clears PSU registers and resets to default settings.
- int **ReportErrors** (int nestNum, int testNum, char errmsg[ERRLEN])
- void **SetPSUName** (char *Name)

Variables

- char **psuName** [25] = {0}
- char **projectDir** [MAX_PATHNAME_LEN] = {0}

2.1.1 Detailed Description

Serial communication wrapper for Ametek used in RCM8 Run-In-Tester.

Author

Dwayne Alex

Copyright

Arxtron Technologies Inc. All Rights Reserved.

Date

8/1/2019 3:38:29 PM

A longer description

Version	Date	Author	Description
1.0.0	Aug 1, 2019	Dwayne Alex	Initial Release
1.0.1	Nov 9, 2020	Jai Prajapati	Updated with library format

2.1.2 Function Documentation

2.1.2.1 GetCurr()

```
double GetCurr (
    char errmsg[ERRLEN] )
```

Get current output level.

/return Double value of current.

2.1.2.2 GetStatus_ERRs()

```
int GetStatus_ERRs (
    char errmsg[ERRLEN] )
```

Gets the error status.

This keeps a queue of the last 10 errors to occur. The error list is extensive, refer to the M9 Programming Manual, section 3.2.5 Error/Event Queue. The ClearPSU() function is used to clear all errors.

2.1.2.3 GetStatus_ESR()

```
int GetStatus_ESR (
    char errmsg[ERRLEN] )
```

Get Event status register.

/return error code listed below Possible vals: 1 - Operation Complete 2 - Request control - not used 4 - Query Error 8 - Device Dependent Error 10 - Execution error 20 - Command error 40- User Request - not used 80 - Power On

2.1.2.4 GetStatus_OUT()

```
int GetStatus_OUT (
    char errmsg[ERRLEN] )
```

Get protection fault status.

Returns

1 - Output ON. 0 - Output off.

2.1.2.5 GetStatus_PROT()

```
int GetStatus_PROT (
    char errmsg[ERRLEN] )
```

Get protection fault status.

/return error code listed below Possible vals: 1 - Constant Voltage 2 - Constant current 4 - Not used 8 - Over voltage protection tripped 10 - Overtemperature protection tripped 20 - Supply external shutdown active 40- Foldback mode operation 80 - Remote programming error

2.1.2.6 GetStatus_SCPI()

```
int GetStatus_SCPI (
    char errmsg[ERRLEN] )
```

Get SCPI status.

/return error code listed below Possible vals: 1 - Not used 2 - PROT Event 4 - error/event queue message available 8 - Questionable Status 10 - Message available 20 - Summary bit for ESR 40 - Request service bit 80 - Operational Status

2.1.2.7 GetStatus_TRIP()

```
double GetStatus_TRIP (
    char errmsg[ERRLEN] )
```

Get tripped status.

Returns

0 - Not tripped. 1 - Tripped.

2.1.2.8 GetVoltage()

```
double GetVoltage (
    char errmsg[ERRLEN] )
```

Get voltage output level.

Returns

Double value of voltage.

2.1.2.9 SelfTest()

```
int SelfTest (
    char errmsg[ERRLEN] )
```

Let the PSU Self-Check.

Returns

0 - No errors, 1 - Error(s) occurred

2.1.2.10 SetDelay()

```
int SetDelay (
    double Time,
    char errmsg[ERRLEN] )
```

Sets the delay for fault reporting on the PSU from the paramter given.

Parameters

in	<i>Time</i>	time in seconds
----	-------------	-----------------

2.1.2.11 SetFold()

```
int SetFold (
    int Type,
    char errmsg[ERRLEN] )
```

Sets the protection type on the PSU from the paramter given.

The control type is fold, there are three modes: type = 0: Do nothing type = 1: Program to down to zero volts upon entering constant voltage mode type = 2: Program down to zero upon entering constant current mode.

2.1.2.12 SetIsolation()

```
int SetIsolation (
    int Iso,
    char errmsg[ERRLEN] )
```

Sets the isolation relay control signal on the PSU from the parameter given.

Parameters

in	<i>Iso</i>	1 - ON, 2 - OFF
----	------------	-----------------

2.1.2.13 SetSense()

```
int SetSense (
    int Sense,
    char errmsg[ERRLEN] )
```

Sets the sense relay signal open/closed on the PSU from the paramter given.

Parameters

in	Sense	1 - ON, 2 - OFF
----	-------	-----------------

2.1.2.14 SetState()

```
int SetState (
    int State,
    char errmsg[ERRLEN] )
```

Sets the output on the PSU from the paramter given.

Parameters

in	State	1 - ON, 2 - OFF
----	-------	-----------------

2.2 C:/Users/jai_prajapati/Documents/SourceLibraries/Serial_LIB/↵ Ametek_LIB/AMETEK_LIB.h File Reference

```
#include <ansi_c.h>
#include "SerialComm_LIB.h"
#include "ArxtronToolslib.h"
```

Functions

- int [Initialize_AMETEK_LIB](#) (char errmsg[ERRLEN])
Initialize Ametek library. Requires SerialComm_LIB to be previously initialized and configured.
- void **GetStandardErrMsg** (int error, char errmsg[ERRLEN])
- int CVICALLBACK **FunctionSelect** (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
- int CVICALLBACK **RunFunction** (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)

- int [GetStatus_ESR](#) (char errmsg[ERRLEN])
Get Event status register.
- int [GetStatus_SCPI](#) (char errmsg[ERRLEN])
Get SCPI status.
- int [GetStatus_PROT](#) (char errmsg[ERRLEN])
Get protection fault status.
- int [GetStatus_ERRs](#) (char errmsg[ERRLEN])
Gets the error status.
- int [GetStatus_OUT](#) (char errmsg[ERRLEN])
Get protection falut status.
- double [GetStatus_TRIP](#) (char errmsg[ERRLEN])
Get tripped status.
- double [GetVoltage](#) (char errmsg[ERRLEN])
Get voltage output level.
- double [GetCurr](#) (char errmsg[ERRLEN])
Get current output level.
- int [SetVolt](#) (double Volts, char errmsg[ERRLEN])
Sets the voltage on the PSU from the paramter given.
- int [SetLimit_Curr](#) (double Current, char errmsg[ERRLEN])
Sets the current limit on the PSU from the paramter given.
- int [SetFold](#) (int Type, char errmsg[ERRLEN])
Sets the protection type on the PSU from the paramter given.
- int [SetPolarity](#) (int Pol, char errmsg[ERRLEN])
Sets the polarity of the PSU from the paramter given.
- int [SetSense](#) (int Sense, char errmsg[ERRLEN])
Sets the sense relay signal open/closed on the PSU from the paramter given.
- int [SetState](#) (int State, char errmsg[ERRLEN])
Sets the output on the PSU from the paramter given.
- int [SetIsolation](#) (int Iso, char errmsg[ERRLEN])
Sets the isolation relay control signal on the PSU from the parameter given.
- int [SetDelay](#) (double Time, char errmsg[ERRLEN])
Sets the delay for fault reporting on the PSU from the paramter given.
- int [InitPSU](#) (double Volts, double Curr, char errmsg[ERRLEN])
Intializes the PSU based off the inputs.
- int [SelfTest](#) (char errmsg[ERRLEN])
Let the PSU Self-Check.
- int [ClearPSUStatus](#) (char errmsg[ERRLEN])
Clears PSU registers.
- int [ResetPSU](#) (char errmsg[ERRLEN])
Clears PSU registers and resets to default settings.
- int **ReportErrors** (int NestNum, int TestNum, char errmsg[ERRLEN])
- void **SetPSUName** (char *Name)

2.2.1 Detailed Description

Author

Dwayne Alex

Copyright

Arxtron Technologies Inc.. All Rights Reserved.

Date

8/1/2019 3:38:29 PM

2.2.2 Function Documentation**2.2.2.1 GetCurr()**

```
double GetCurr (
    char errmsg[ERRLEN] )
```

Get current output level.

/return Double value of current.

2.2.2.2 GetStatus_ERRs()

```
int GetStatus_ERRs (
    char errmsg[ERRLEN] )
```

Gets the error status.

This keeps a queue of the last 10 errors to occur. The error list is extensive, refer to the M9 Programming Manual, section 3.2.5 Error/Event Queue. The ClearPSU() function is used to clear all errors.

2.2.2.3 GetStatus_ESR()

```
int GetStatus_ESR (
    char errmsg[ERRLEN] )
```

Get Event status register.

/return error code listed below Possible vals: 1 - Operation Complete 2 - Request control - not used 4 - Query Error 8 - Device Dependent Error 10 - Execution error 20 - Command error 40- User Request - not used 80 - Power On

2.2.2.4 GetStatus_OUT()

```
int GetStatus_OUT (
    char errmsg[ERRLEN] )
```

Get protection falut status.

Returns

1 - Output ON. 0 - Output off.

2.2.2.5 GetStatus_PROT()

```
int GetStatus_PROT (
    char errmsg[ERRLEN] )
```

Get protection fault status.

/return error code listed below Possible vals: 1 - Constant Voltage 2 - Constant current 4 - Not used 8 - Over voltage protection tripped 10 - Overtemperature protection tripped 20 - Supply external shutdown active 40- Foldback mode operation 80 - Remote programming error

2.2.2.6 GetStatus_SCPI()

```
int GetStatus_SCPI (
    char errmsg[ERRLEN] )
```

Get SCPI status.

/return error code listed below Possible vals: 1 - Not used 2 - PROT Event 4 - error/event queue message available 8 - Questionable Status 10 - Message available 20 - Summary bit for ESR 40 - Request service bit 80 - Operational Status

2.2.2.7 GetStatus_TRIP()

```
double GetStatus_TRIP (
    char errmsg[ERRLEN] )
```

Get tripped status.

Returns

0 - Not tripped. 1 - Tripped.

2.2.2.8 GetVoltage()

```
double GetVoltage (
    char errmsg[ERRLEN] )
```

Get voltage output level.

Returns

Double value of voltage.

2.2.2.9 SelfTest()

```
int SelfTest (
    char errmsg[ERRLEN] )
```

Let the PSU Self-Check.

Returns

0 - No errors, 1 - Error(s) occurred

2.2.2.10 SetDelay()

```
int SetDelay (
    double Time,
    char errmsg[ERRLEN] )
```

Sets the delay for fault reporting on the PSU from the paramter given.

Parameters

in	<i>Time</i>	time in seconds
----	-------------	-----------------

2.2.2.11 SetFold()

```
int SetFold (
    int Type,
    char errmsg[ERRLEN] )
```

Sets the protection type on the PSU from the paramter given.

The control type is fold, there are three modes: type = 0: Do nothing type = 1: Program to down to zero volts upon entering constant voltage mode type = 2: Program down to zero upon entering constant current mode.

2.2.2.12 SetIsolation()

```
int SetIsolation (
    int Iso,
    char errmsg[ERRLEN] )
```

Sets the isolation relay control signal on the PSU from the parameter given.

Parameters

in	<i>Iso</i>	1 - ON, 2 - OFF
----	------------	-----------------

2.2.2.13 SetSense()

```
int SetSense (
    int Sense,
    char errmsg[ERRLEN] )
```

Sets the sense relay signal open/closed on the PSU from the paramter given.

Parameters

in	<i>Sense</i>	1 - ON, 2 - OFF
----	--------------	-----------------

2.2.2.14 SetState()

```
int SetState (
    int State,
    char errmsg[ERRLEN] )
```

Sets the output on the PSU from the paramter given.

Parameters

in	<i>State</i>	1 - ON, 2 - OFF
----	--------------	-----------------

Index

AMETEK_LIB.c

- GetCurr, [5](#)
- GetStatus_ERRs, [5](#)
- GetStatus_ESR, [5](#)
- GetStatus_OUT, [5](#)
- GetStatus_PROT, [5](#)
- GetStatus_SCPI, [6](#)
- GetStatus_TRIP, [6](#)
- GetVoltage, [6](#)
- SelfTest, [6](#)
- SetDelay, [7](#)
- SetFold, [7](#)
- SetIsolation, [7](#)
- SetSense, [8](#)
- SetState, [8](#)

AMETEK_LIB.h

- GetCurr, [10](#)
- GetStatus_ERRs, [10](#)
- GetStatus_ESR, [10](#)
- GetStatus_OUT, [10](#)
- GetStatus_PROT, [10](#)
- GetStatus_SCPI, [11](#)
- GetStatus_TRIP, [11](#)
- GetVoltage, [11](#)
- SelfTest, [11](#)
- SetDelay, [12](#)
- SetFold, [12](#)
- SetIsolation, [12](#)
- SetSense, [13](#)
- SetState, [13](#)

AMETEK_LIB.h, [10](#)

- GetStatus_SCPI
 - AMETEK_LIB.c, [6](#)
 - AMETEK_LIB.h, [11](#)
- GetStatus_TRIP
 - AMETEK_LIB.c, [6](#)
 - AMETEK_LIB.h, [11](#)
- GetVoltage
 - AMETEK_LIB.c, [6](#)
 - AMETEK_LIB.h, [11](#)

- SelfTest
 - AMETEK_LIB.c, [6](#)
 - AMETEK_LIB.h, [11](#)

- SetDelay
 - AMETEK_LIB.c, [7](#)
 - AMETEK_LIB.h, [12](#)

- SetFold
 - AMETEK_LIB.c, [7](#)
 - AMETEK_LIB.h, [12](#)

- SetIsolation
 - AMETEK_LIB.c, [7](#)
 - AMETEK_LIB.h, [12](#)

- SetSense
 - AMETEK_LIB.c, [8](#)
 - AMETEK_LIB.h, [13](#)

- SetState
 - AMETEK_LIB.c, [8](#)
 - AMETEK_LIB.h, [13](#)

C:/Users/jai_prajapati/Documents/SourceLibraries/Serial_LIB/Ametek_LIB/AMETEK_LIB.c,
[3](#)

C:/Users/jai_prajapati/Documents/SourceLibraries/Serial_LIB/Ametek_LIB/AMETEK_LIB.h,
[8](#)

GetCurr

- AMETEK_LIB.c, [5](#)
- AMETEK_LIB.h, [10](#)

GetStatus_ERRs

- AMETEK_LIB.c, [5](#)
- AMETEK_LIB.h, [10](#)

GetStatus_ESR

- AMETEK_LIB.c, [5](#)
- AMETEK_LIB.h, [10](#)

GetStatus_OUT

- AMETEK_LIB.c, [5](#)
- AMETEK_LIB.h, [10](#)

GetStatus_PROT

- AMETEK_LIB.c, [5](#)