
Triangle MicroWorks, Inc.
IEC 60870-5-102 Slave
Source Code Library

What's New?

a description of the changes included in all versions of the library

Version 3.13.00
November 17, 2011

Property of Triangle MicroWorks, Inc.



This Source Code and the associated Documentation contain proprietary information of Triangle MicroWorks, Inc. and may not be copied or distributed in any form without the written permission of Triangle MicroWorks, Inc.

Copies of the source code may be made only for backup purposes.





© 1994 - 2011 Triangle MicroWorks, Inc. All rights reserved.

This document describes features or corrections that have been added to the IEC 60870-5-101 Slave Source Code Library.





The symbols to the left of each revision are used to help define the following kinds of revisions:


- 101** Additional or enhanced IEC 60870-5-101 Features.
-  Additional Implementation Features, allowing implementers to more efficiently install the Source Code Library.
-  Corrections to problems, with indications for when the problems were introduced.

Version 3.13.00 (November 17, 2011)

-  Modified 56-bit time decoding so that years 70-99 assume 1970-1999 instead of 2070-2099
-  Corrected a warning caused by an uninitialized structure in LinIOTarg RS232 implementation.
-  Provided big endian get and put functions.
-  *Description:* LinIOTarg could access channel structure members after marking a channel for deletion.
Introduced: v3.07
Resolution: Fixed in v3.13


Version 3.12.00 (July 7, 2011)

-  Added TMWTYPES_ANALOG_TYPE_DSCALED that lets database return Double and Long value. This feature provides more flexibility to the outstation for handling rounding.
-  *Description:* The timer callback may continue to loop if a very short timer is added in the timer callback.
Introduced: v3.00
Resolution: Fixed in v3.12
-  *Description:* tmwdtime_addOffset did not follow 0 -599999 rule.
Introduced: v3.00
Resolution: Fixed in v3.12
-  *Description:* Some compilers issued warnings when a break followed a return (e.g., in tmwphysd.c).
Introduced: v3.00
Resolution: Fixed in v3.12

- 


Description: In function `tmwmem_lowlnit()`, a failure to allocate memory resulted in a return with no indication to the caller that a failure occurred.

Introduced: v3.00

Resolution: Fixed in v3.12
- 

Description: The link layer incorrectly assumed that unconfirmed data was a broadcast message.

Introduced: v3.00

Resolution: Fixed in v3.12
- 

Description: `LinIOTarg` computed date incorrectly when converting a “struct tm” to `TMWDTIME`.

Introduced: v3.07

Resolution: Fixed in v3.12


Version 3.11.00 (October 1, 2010)


No updates to IEC 60870-5-102 Slave Source Code Library in this release.


Version 3.10.00 (August 15, 2010)


No updates to IEC 60870-5-102 Slave Source Code Library in this release.


Version 3.09.00 (March 15, 2010)

- 

Modified `WinIOTarg` to prefer an exact address match when multiple channels are open and `disconnectOnNewSyn==true`.
- 


Added `localIpAddress` to `WinIoTarg` to specify the source address to use when Client (or UDP sender) is sending IP messages. This, in conjunction with the Destination IP address and Routing Tables allows control over which Ethernet card is used to establish a connection. (At Windows command prompt, enter "route print" "route add xxx" can be used to add a route to cause a particular NIC to be used.)
- 

Enhanced `WinIOTarg` to support a list of addresses from which TCP/IP connections will be accepted.
- 

Added user data pointers to `TMWCHNL`, `TMWSCTR`, and `TMWSESN`.
- 

Description: When multiple application contexts are used, closing one deleted the memory pool lock, which could cause problems if another application context was open.


Introduced: v3.00

Resolution: Fixed in v3.09
- 

Description: `WinIOTarg` could fail to process all available characters, causing timeouts when running at high baud rates.


Introduced: v3.00

Resolution: Fixed in v3.09

 *Description:* When configured to use system time, WinIOTarg System time not setting the dstInEffect flag correctly.

Introduced: v3.00


Resolution: Fixed in v3.09

 *Description:* LinIOTarg could cause high levels of processor utilization when multiple channels were opened.

Introduced: v3.07

Resolution: Fixed in v3.09

Version 3.08.00 (September 25, 2009)

 *Description:* Modifying IEC 60870-5-102 Common Address of ASDU Size did not work.

Introduced: v3.00


Resolution: Fixed in v3.08


 *Description:* The rs232Reader thread in WinIOTarg was missing a stop event.


Introduced: v3.00

Resolution: Fixed in v3.08


Version 3.07.00 (July 17, 2009)

 Added LinIOTarg (low-level target for Linux) and new command line-based examples (that can be used with Linux).

 Integrated WinIOTarg (low-level target for Windows) into the standard release package.

 Added FormatMessage Win32 API function to WinIOTarg to provide more descriptive errors instead of just numeric values.

Version 3.06.00 (April 14, 2009)

 Added TMWCNFG_MAX_APPLRCVS to “break out” of loop in tmwappl_checkForInput() after the specified number of iterations.

 Modified timer values in tmwlink_channelCallback to speed up retries.

Version 3.05.00 (December 4, 2008)

No updates to IEC 60870-5-102 Slave Source Code Library in this release.

Version 3.04.01 (September 30, 2008)

 Improved comments on compiler options such as TMWCONFIG_USE_MANAGED_SCL.

Version 3.04 (September 12, 2008)



Description: tmwdb_lockQueue and tmwdb_unlockQueue were described in the header comments for tmwdb_storeEntry in tmwdb.h, but they were not defined in tmwdb.c.

Introduced: v3.00

Resolution: Fixed in v3.04

Version 3.03.00 (April 1, 2008)



Added ability to include session indicator in channel statistics.

Version 3.02.00 (December 7, 2007)

No updates to IEC 60870-5-102 Slave Source Code Library in this release.

Version 3.01.01 (August 8, 2007)



Added ability to keep time on a per-session basis. Added pSession to tmwdtime structure; target layer can use this parameter to return a time specific to the session.

Version 3.01.00 (May 23 2007)



Initial release of IEC 60870-5-102 v3.x Source Code Library

The Triangle MicroWorks, Inc. Source Code architecture was redesigned and reimplemented to follow modern software engineering practices and leverage new techniques in software design. We have also incorporated suggestions from our existing customer base. The main advantages to this redesign include:

- 1) Common source code architecture across all TMW libraries - The use of a common architecture across all libraries significantly reduces time required to port additional protocols to a target device. In addition, using common software across all products results in better code, since common functions are utilized much more frequently than protocol specific functions.
- 2) Source Code Library calling routines are now compatible with a wider variety of event driven techniques, allowing the target application to be achieve higher performance.
- 3) Most configuration parameters are now passed as arguments in Source Code Library function calls instead of macros. The allows for data hiding and protects configuration parameters from accidentally being changed while the Source Code Library is running. In addition, macro calls to target hardware and the database interface routines were replaced with function calls.

- 4) Significant reduction in the time to implement new features - In addition to being more flexible, the use of modern software design practices significantly reduces the time required for Triangle MicroWorks to support new features. As the existing protocols are constantly being improved by the associated Technical Committees and Working Groups, it is essential that the Source Code Libraries are able to remain up to date with the latest standards.

Version 2.13 (July 10, 2002):



Removed `S101CNFG_LINK_SUPPORT_BALANCED()` and `S101CNFG_LINK_SUPPORT_UNBALANCED()` from `s101cnfg.h`. Now both are supported at all times.

Version 2.12 (March 21, 2002):

- 101 Added support for protection equipment events including ASDU types 17(MEPTA), 18(MEPTB), 19(MEPTC), 38(MEPTD), 39(MEPTD), and 40(MEPTF).

Version 2.11 (January 24, 2002):

- 101 Added support for file transfer in the control and monitor direction.



Fixed a bug that caused the slave SCL to deadlock when running in balanced mode if the master restarted during an unsolicited response from the slave.

Version 2.10 (December 7, 2001):



Fixed a bug that caused the Source Code Library to hang when we received multiple simultaneous requests in balanced mode. We now wait for any data in the physical transmit buffer to be sent before checking for received messages. We do this since we do not currently buffer the link layer confirm and sending it would overwrite the data currently in the transmit buffer.



Fixed a bug that caused immediate retries if the message was not transmitted immediately (i.e. `S101TARG_COMM_TRANSMIT_READY` macro returns false and/or `S101CNFG_PHYS_TX_DLY` greater than 0). The link layer would incorrectly assume the confirm timeout has expired and try to retransmit the message immediately.



Fixed a bug that was causing an incorrect value for Cause of Initialization to be returned in the End Of Initialization message.



Fixed a bug that was causing an invalid sector index to be passed to the `S101DATA_CLEAR_SECTOR_RESET`.

Version 2.09 (November 9, 2001):

- 101 Added support for custom configuration table ASDU types (MCTNA and CCTNA).

101 Redesigned initialization for balanced mode to be more robust and support inhibiting user data transmission until both sides of the communications channel have been initialized.

101 Return Directory ASDU (126) message with 0 files in response to Call Directory request.



Fixed a bug that would attempt to send messages longer than the transmit buffer size if time tags were enabled in GI, Counter Interrogation, and Read responses. Introduced in version 2.07.

Version 2.08 (October 17, 2001):

101 Added functionality that will reinitialize a session when we receive a link reset from the remote device if user data was received since the last link reset was issued. This allows us to properly re-establish communications with a master that has restarted.

101 Added new S101CNFG_LINK_RESET_EOI parameter to allow the generation of an End Of Initialization message whenever we receive a link reset.



Fixed compile errors that occurred if S101DATA_SUPPORT_MITNA was set to TMWDEFS_FALSE. Introduced in version 2.06.



Offline poll delay was incorrectly being used for unbalanced slaves. Introduced in version 2.07.



Request Status Of Link was not being generated for slave operating in balanced mode. This bug only occurred if S101CNFG_SUPPORT_UNBALANCED was set to TMWDEFS_FALSE. Introduced in version 2.07.

Version 2.07 (October 5, 2001):

101 Added point specific time format options for General Interrogation, Counter Interrogation and Read response as well as events.



Added initialization of **firstCharWait** timer and changed call from **S101TARG_COMM_CLOSE()** to **tmwphys_closeChannel()** to properly close channels. Introduced in version 2.02.



Fixed bug with **S101CNFG_PHYS_TX_DLY()** which would not allow transmission if this macro was defined as zero. Introduced in version 2.02.



The **DIR** bit is no longer set for unbalanced mode. Introduced in version 2.02.










Version 2.06 (September 4, 2001):

101 Added limited file ID support (Type ID's 122 and 123).



Fixed compile time errors when **S101CNFG_LINK_SUPPORT_UNBALANCED()** is set false or when **S101TARG_DIAG_FEATURE_MASK()** is set to zero. Introduced in v2.02.

Version 2.05 (August 31, 2001):




- 101 Added new macros: **S101TARG_MEMCOPY()** to specify local memory copy implementation, **S101DATA_SESSION_NOTIFY_ONLINE()** to support target notification of online/offline events. **S101TARG_MALLOC()** and **S101TARG_FREE()** to support dynamic allocation of memory.
- 101 Added support for Bitstring commands (Type ID 51) and Bitstring data (Type ID 7, 8 and 33) with and without time tags.
- 101 Added support for custom ASDU types: 240 – Integrated Totals BCD, 241 – Integrated Totals BDC with Time Tag and 242 – Integrated Totals BDC Command.
-  Removed unused declaration of **s101dbas_mstnaUseSequence**. The step position information (**MSTNA**) uses **s101dbas_neverUseSequence()**.
-  Added new doubly linked list utilities in **tmwdb1nk.c** and **tmwdb1nk.h**.
-  Variable structure qualifier definitions have been moved from **I870def1.h** to **I870defs.h**. Replaced **I870DEF1_VSQ_SEQUENCE_ADDR_MASK** with **I870DEFS_VSQ_SQ**.
-  Added return status to **i8701nk1_ackSentSequence()**. This status will return false if the channel was closed because of an error, otherwise it returns true for success.
-  Renamed **tmwphys_initChannel()** to **tmwphys_openChannel()** and moved **S101TARG_COMM_OPEN()** macro to inside of **tmwphys_openChannel()**.
-  Added **TMWSTAT_LINK_UNEXPECTED_ACK**, an unexpected acknowledge error code and **TMWSTAT_LINK_REQUEST_CANCELED**, a link level request was cancelled and the session was restarted error code.
-  Changed application callback functions to be accessed through a structure to reduce the number of calling parameters to **I870SCL_BUILD_OUTGOING_FUNC**. Added **beforeTxCallback** and **callbackParam** to application callback capabilities. Defined new **restartSession** callback used to tell the application layer that a link layer session restart occurred.
-  Changed parameters to **I870SCL_PROCESS_INCOMING_FUNC** and **s101appl_processRequest()**.
-  Corrected a problem with **tmwc1prm_parseSector()** that could have resulted in illegal references of memory when multiple Source Code Libraries are installed. Introduced in version 2.02.

Version 2.02 (March 14, 2001):

- 101 IEC 60870-5-101 Source Code Library can be installed in combination with IEC 60870-5-104 Source Code Library using many of the same application layer functions to save code space.

- 101 2 byte Cause of Transmission with Originator Address is supported.
- 101 Added support for ASDU types 13, 14, 36, 50, and 112 (short floating point measurands, set point commands, and parameters).
- 101 Now, for single point information objects, double-point information objects, and step position information objects, the cause-of-transmission is stored with change events in change event buffers: a new **eventCOT** parameter has been added to **s101rbe_mspnaStoreChangeEvent()**, **s101rbe_mdpnaStoreChangeEvent()**, **s101rbe_mstnaStoreChangeEvent()**. This allows changes resulting from control operations to be transmitted using proper cause-of-transmission (11 or 12). Previously, only a cause-of-transmission value of 3 would have been used, and they were always non-time stamped.
- 101 Now, when a single point, double point or step point control is finished, the Source Code Library will automatically generate a cause-of-transmission 11 event for the monitored point, but only if the Target Application did not call the corresponding **s101rbe_mspnaStoreChangeEvent()**, **s101rbe_mdpnaStoreChangeEvent()**, **s101rbe_mstnaStoreChangeEvent()** for that monitored point function since the control operation was initiated.
- 101 Now, if cause-of-transmission 11 or 12 events exist for a monitored point corresponding to a control operation, then all events buffered before and including the cause-of-transmission 11 or 12 event will be transmitted before the ACTTERM is transmitted for the control operation. The specific new behavior is that it is now possible for change events using any cause-of-transmission to be transmitted while an ACTTERM is pending.

New files and changed filenames:

-  Created **s101cmd.c** and **s101cmd.h** to define functions for a remote terminal command line interface. Created new **s101sc1.h** to consolidate literal constant and data type definitions. Created **s101cmts.c** and **s101cmts.h** to define command line interface user test commands.
-  If an IEC 60870-5 Slave Source Code Library supporting multiple protocols is purchased, the following file name prefixes will be “s870”. Otherwise the prefix specifies the supported protocol for example “s101”, “s102” etc. For the IEC 60870-5-101 Slave Source Code Library, the files are as follows: **s101app1.c**, **s101app1.h**, **s101cmd.c**, **s101cmd.h**, **s101cmts.c**, **s101cmts.h**, **s101cnfg.h**, **s101data.h**, **s101dbas.c**, **s101dbas.h**, **s101rbe.c**, **s101rbe.h**, **s101sc1.h**, and **s101targ.h**.
-  Created **tmwphys.c** and **tmwphys.h** to define physical layer functions used to receive and transmit. Created **tmwscript.c** and **tmwscript.h** to define diagnostic memory commands for executing scripts. Created **tmwclprm.c** and **tmwclprm.h** to define command line parameter parsing functions. Created **tmwclraw.c** and **tmwclraw.h** to define the command line interface transmit raw command. Created

tmwstrf1.c and **tmwstrf1.h** to define how to convert to/from floating point representations in strings. Created **tmwan1zr.c** and **tmwan1zr.h** to define protocol analyzer functions common for all Source Code Libraries. Created **tmwomni.c** and **tmwomni.h** to define definitions and functions used when more than one TMW Source Code Library is compiled and linked together in an effort to save code space (all SCL's can be compiled and linked separately without **TMWOMNI** if desired). Created **tmwstat.h** to define status and error codes for all TMW Source Code Libraries. All the above modules are common to all IEC 60870 Source Code Libraries.



Created **pcscript.h** to define scripts to be included in PC-based user interfaces.



Created **an1zbuf.h** to define a class which holds one diagnostic/protocol analyzer string.



Created **rtarg.h** to handle remote terminal command line params.



Renamed **strescap.c** to **tmwstres.c** and **strescap.h** to **tmwstres.h** to handle string escape sequence functions.



Renamed **i1slink.c** to **i870lnk1.c** and **i1slink.h** to **i870lnk1.h**. Encapsulated all link layer code in these modules by moving code from **s101dvr.c** and **s101task.c**. These files are not specific to the slave IEC 60870-5-101 Source Code Library, and may be included with IEC 60870-5-101 Master Source Code Library.



Created new **i870sc1.h** which includes definitions common to all IEC 60870 Source Code Libraries.



Created **i870an11.c** and **i870an11.h** to define protocol analyzer functions for IEC 60870-5-101 link layer frames.



Created **i870defs.h** and **i870def1.h** from **i101defs.h**. **i870defs.h** contains literal constant and data type definitions that are specific to all IEC 60870-5-10x protocols. **i870def1.h** contains literal constant and data type definitions that are specific to the IEC 60870-5-101 protocol. These files are not specific to the slave IEC 60870-5-101 Source Code Library, and may be included with IEC 60870-5-101 Master Source Code Library, or other IEC 60870-5-10x Source Code Libraries.






Renamed **ieCft12.h** and **ieCdtim.*** to **i870ft12.h** and **i870dtim.***, respectively, to consolidate all protocol-specific files into a common prefix: **i870**. Like the **i870defs.h** and **i870def1.h** files, these files are not specific to the slave IEC 60870-5-101 Source Code Library, and may be included other IEC 60870-5-10x Source Code Libraries.











Renamed **i1ssim.*** to **simdata.*** (and symbols defined within them) to more clearly indicate that these files are example files, and not normally included in final implementations of the Source Code Library. Also, cleaned up **s101data.h**, and examples within **simdata.*** to avoid confusion that the examples must be explicitly followed in final implementations of **s101data.h**.

TA-to-SCL interface changes:

-  Now passes `clientIndex` to `s101task_main()` (formerly called `i1sdvrs_process101()`) instead of a pointer to a `I1STASK_CLIENT_INFO` structure (a pointer to a member of the `i1sdvrs_clientInfo[]` array). The Target Application no longer needs to include knowledge of the `i1sdvrs_clientInfo[]` structure array in its interface with the Source Code Library.
-  `s101rbe_mxxxxStoreChangeEvent()` now accepts `TMWDEFS_NULL` for `pTimeStamp`. If `TMWDEFS_NULL` is passed, the current system time will be used.
-  To aid understanding and ease implementation, `TMWDEFS_COI` (an enumerated data type) is passed instead of `I870DEF1_COI` (a bitmask field) to the `s101task_init()` and `s101task_close()`.

New Target Hardware Interface macros:

-  Added `S101TARG_COMM_MAX_NUM_CHNLS()` and `S101TARG_COMM_NUM_ACTIVE_CHNLS()`. The former is a compile-time constant which sets the size of statically allocated arrays; the latter is a run-time constant which sets the number of communication channels open at any one particular time.
-  Added `S101TARG_MALLOC()` and `S101TARG_FREE()` macros to support dynamic memory allocation of change event buffers.
-  Changed `S101TARG_DIAG_AUX_PUT_STRING()` to use a pointer to a `TMWDEFS_ANLZ_ID` structure instead of the `S101ANLZ_ID` structure.
-  Added `S101TARG_RESOURCE_LOCK()` and `S101TARG_RESOURCE_UNLOCK()` to support multiple thread access to event buffers.
-  Added `S101TARG_DIAG_COMMANDS()` so that target hardware specific commands can be added to the command line interface.
-  Added `S101TARG_UNUSED_PARAMETER()` to help prevent compiler warnings for some strict compilers.
-  Added `S101TARG_xxxx_INSTALLED`, which are used if multiple TMW Source Code Libraries are installed. Moved `S101CNFG_DIAG_FEATURE_MASK` to `S101TARG_DIAG_FEATURE_MASK` so that initial phases of installation will not require editing of `s101cnfg.h`. Added `S101TARG_ANLZ_xxxx` macros to replace macros in `s101anlz.h`. These macros allow target application customization of protocol analyzer displays.
-  Removed `interTime` as a parameter to the `S101TARG_COMM_RECEIVE()`. Target applications should use their own timeout value to determine if an inter-character timeout has occurred. This timeout may be set much tighter, since it may be used by very low-level communications routines, than the

S101CNFG_PHYS_CHAR_TIMEOUT(), which is used by the source code library to perform similar but not as precise inter-character timeout tests.



Moved **S101CNFG_CLOCK_VALID_PRD** to the module **s101targ.h** and renamed it **S101TARG_CLOCK_VALID_PRD()**.



Added **S101TARG_COMM_CHNL_NAME()** to allow diagnostic or protocol analyzer display functions to indicate applicable communication channels in a more easily understood fashion.

Database Interface changes:



Renamed the **causeOfInit** parameters to **coiField** for **S101DATA_ANY_SECTOR_HAS_RESET()** and **S101DATA_RESET_SECTOR()** since those parameters still are of **I870DEF1_COI** type.



Added **S101DATA_EXECUTE_CRPNA_QRP()** which allows target application total control over the reset process command. Renamed **S101DATA_ALLOW_CRPNA_QRP_xxx()** to **S101DATA_ALLOW_CRPNA_QRP()** and now pass the qualifier of the reset process command to it.



Can now save code space by removing support for time tagged events and unbalanced link mode using the following support #defines:
S101CNFG_LINK_SUPPORT_UNBALANCD, **S101DATA_SUPPORT_CSCTA**,
S101DATA_SUPPORT_CDCTA, **S101DATA_SUPPORT_CRCTA**,
S101DATA_SUPPORT_CSETA, **S101DATA_SUPPORT_CSETB** and
S101DATA_SUPPORT_CSETC.



IEC 60870-5-101 Slave Source Code Library supports IEC 60870-5-104 Slave Source Code Library data types with 56 bit time tags.



Added time-tagged control types defined for IEC 60870-5-104. These types are included in the IEC 60870-5-101 library to simplify translation in protocol conversions. Added selection for these types with **S101DATA_SUPPORT_CSCTA**, **S101DATA_SUPPORT_CDCTA**, **S101DATA_SUPPORT_CRCTA**, **S101DATA_SUPPORT_CSETA**, **S101DATA_SUPPORT_CSETB**, and **S101DATA_SUPPORT_CSETC**.











During a select/execute sequence using time-tagged control types, the time tags are not compared and therefore do not have to be equal.






Added **S101DATA_SUPPORT_CRDNA**, **S101DATA_SUPPORT_CCSNA**, **S101DATA_SUPPORT_CTSNA** and **S101DATA_SUPPORT_CCDNA**.



Replaced **S101DATA_xxxxx_GROUP_OK()** with **S101DATA_xxxxx_GROUP_MASK()** which now returns a mask instead of evaluating a mask. Added **pMmenxTableIndex** parameter to **S101DATA_PMENX_READ()**. Created **S101DATA_Cxxxx_CTRL_MASK()** and replaced **S101DATA_Cxxxx_STORE()** with **S101DATA_Cxxxx_SELECT()** and **S101DATA_Cxxxx_CONTROL()**.


-  Changed default behavior of `S101DATA_xxx_SECTOR_DATE_TIME()` to call `s101task` functions.
-  Created `S101DATA_DIAG_COMMANDS()` macro so that database-specific commands can be added to the command line interface and added example `simdata` commands.
-  Now passes `pPointOrigin` to `simdata_cxxxstore()` so that error messages can be displayed in the protocol analyzer if a control is not successful.
-  Added support for floating point measurands, commands and parameters (`MMENC`, `MMETC`, `MMETF`, `CSENC`, `FMENC`).
-  Combined `S101DATA_Mxxxx_READ()` and `S101DATA_Mxxxx_CHANGED()` macros into a new `S101DATA_Mxxxx_GET_CURRENT_VALUE()` macro. Also created `S101DATA_Mxxxx_GET_LAST_REPORTED()` macros. This, together with the new `S101TARG_RESOURCE_LOCK()` and `S101TARG_RESOURCE_UNLOCK()` macros, ensures that transmitted static data reflects the state of the most recently transmitted change event data.
-  Changed `S101DATA_Cxxxx_Mxxxx_POINT_NUM()` to `S101DATA_Cxxxx_Mxxxx_TABLE_INDEX()` which removed the need for `S101DATA_Cxxxx_Mxxxx_READ()` macros. Changed example implementations of `S101DATA_Cxxxx_Mxxxx_READ` to use `mxxxx` read functions, but also added new `pChangeType` and `ctrlRespPending` parameters to those functions.
-  Removed `S101CNFG_DIAG_ID_AUX_MASK()`, `S101CNFG_DIAG_STATISTIC_COUNTER` (a type definition), `INC_ERROR_COUNTER()` and `S101CNFG_DIAG_STATISTIC_COUNTER()` and replaced them with `S101DATA_NOTIFY_xxxx` macros. This allows Target Application to take any necessary action, including none.
-  Added `S101DATA_NOTIFY_EVENT_OVERFLOW()` to replace `s101rbe_eventBufferOverflowed()`, which was available to be called by the Target Application. The `S101DATA_NOTIFY_EVENT_OVERFLOW()` macro is called by the Source Code Library. The event buffer overflow state and count is no longer maintained by `s101rbe.c` module. Instead, it is up to the Target Application to take appropriate action upon being called through `S101DATA_NOTIFY_EVENT_OVERFLOW()`.


Configuration Interface Changes:


-  Moved `S101RBE_TIME_FORMAT` definitions to `s101sc1.h` and consequently renamed them.
-  Added configuration for the class assignments for `CCINA`, `CICNA`, `CRDNA` and `CTSNA`.
-  Replaced `S101CNFG_DIAG_STRINGS_ENABLED()` with `S101TARG_DIAG_FEATURE_MASK()`.

 Enhanced description of **S101CNFG_PHYS_CHAR_TIMEOUT()** which is now only used internally.

 Changed name of **S101APPL_CLASS_ID** to **S101SCL_CLASS_ID**.

 Added (**clientIndex**, **sectorIndex**) to **S101CNFG_Mxxxx_EVENT_BUFFER_SIZE** configuration parameter because now, depending upon the definition of **S101TARG_MALLOC()**, these parameters can be different for each client and sector and can be determined at run-time. However, static allocation of event buffers is still supported for Target Applications that do not support dynamic memory.

 Created **S101CNFG_DIAG_SCRIPT()** macros which may be used to include pre-built sequences of commands into the command line interface.


 Renamed:


S101CNFG_LINK_BALANCED()	to	S101CNFG_LINK_MODE()
S101CNFG_SUPPORT_BALANCED()	to	S101CNFG_LINK_SUPPORT_BALANCED()
S101CNFG_LINK_TOUT_ACK()	to	S101CNFG_LINK_CNFM_TIMEOUT()
S101CNFG_LINK_RETRIES()	to	S101CNFG_LINK_MAX_RETRIES()
S101CNFG_LINK_DIR()	to	S101CNFG_OPT_BALANCED_DIR_A()
S101CNFG_DIAGPHYS_NUM_CLIENTS()	to	S101TARG_COMM_MAX_NUM_CHNLS()
S101APPL_SIZE_CAUSE_OF_TRANS()	to	S101CNFG_ASDU_SIZE_COT()
S101CNFG_PHYS_NUM_CLIENTS()	to	S101TARG_COMM_MAX_NUM_CHNLS()

 Created **S101CNFG_CLOCK_ACCEPT_TIMESYNC()** and **S101CNFG_APPL_COMMAND_TIMEOUT()**.


 Removed **S101CNFG_DIAG_ID_AUX_MASK()** and **S101CNFG_DIAG_STATISTIC_COUNTER()** which have been replaced by **S101DATA_NOTIFY_xxxx** macros.


Protocol Analyzer Changes:

 Changed format of diagnostic protocol analyzer message to include communication channel name (through new **S101TARG_COMM_CHNL_NAME()** macro) rather than index, and to more closely match the format of other Triangle MicroWorks, Inc. Source Code Library protocol analyzer displays.

 Added new displayable physical layer for the protocol analyzer. Changed the link layer to display link frames only which may duplicate physical bytes if bytes are being reparsed.

 The originator address is displayed from the 2 byte cause of transmission.

 Added protocol analyzer display of transmitted application layer messages.

 Now sends error messages to the protocol analyzer display if control points cannot be found, or in the example **simdata.c**, if control operations are not successful.

 Renamed:

I1SANLYZ_ID	to TMWDEFS_ANLZ_ID
I1SANLYZ_FILTER_XXXX	to TMWDEFS_DIAGID_XXXX
I1SANLYZ_S101LIB_XXXX	to S101ANLYZ_SCL_XXXX



With the removal of **S101CONFIG_DIAG_ID_AUX_MASK()**, the filtering of the diagnostic protocol analyzer messages is now totally up to the Target Application.

General changes:



Added **I870DEF1_QUALITY_OV** to **i870def1.h** for counter quality and added **I870DEF1_ATTRIB_XXX** since they are unique to the IEC 60870-5-101 protocol Source Code Library. Removed some underscores from the **I870DEF1_TYPE_XXXXX** definitions and added **I870DEF1_QOI** type and constants. Added **I870DEF1_QUALITY_EI** and **I870DEF1_DCS_MASK**. Removed **I870DEF1_DCS_INTERMEDIATE** and **I870DEF1_OFFSET_COT**. Revised naming of **I870DEF1_QOC_XXXX** constants. Added **I870DEF1_QOS**, **I870DEF1_QOS**, **I870DEF1_QOS** and **I870DEF1_QOS** type definitions. Added **I870DEF1_COT_UNKNOWN_XXXX**.



Reorganized transmit frame buffer so only one large buffer is required.



Added some type casting to remove warnings from specific strict compilers.



In all symbol names, removed underscore from standard ASDU type abbreviations; for example **M_SP_NA** has been changed to **MSPNA**. This makes it easier to search for uses of ASDU types.



Changed some basic types, added example **#pragma's**, initialized local variables, and added type casts to prevent compiler warnings for some strict compilers.

Bug Corrections:



No longer allow single character frames detected during re-parsing, or after a previous parsing error (they are allowed only after a successful frame has been parsed or after an inter-character timeout).



To avoid a bug difference between MS VC++ 6.0 and MS VC++ 5.0, moved definition of constructor for **NEWARG_BOOL** to inside source file, and removed **boolean_values** as a member of **NEWARG_BOOL**.



Corrected error in **tmwdtime_decrementHour()** in module **tmwdtime.c** that could convert a timestamp with no day of week value into a timestamp with a day of week value of "Sunday" <7>.

Version 1.29 (November 3, 1999):

- 101** Addendum 2 of IEC 870-5-101 now allows class 1 response data when no class 2 data is available. The function **i1s1ink_processReceive()** was modified to support this and to pass user data with confirm messages to the application layer first, so

the ACK response access demand bit reflects any data generated by the message. These enhancements eliminate the need for an extra class 2 data poll when class 1 data is available.

101 Change select command arm timer type from unsigned short to unsigned long. This allows for more than 65 second transmission time on satellite networks.



Fixed problem that prevented ACTCON from being sent in response to execute commands when the immediate response from the `i1sdbas_storeCommandData()` function was `I1SDBAS_COMSTAT_SUCCESSFUL`. It had been assumed that `i1sdbas_storeCommandData()` would return `I1SDBAS_COMSTAT_EXECUTING` at least once. This problem was introduced in version 1.22.

Version 1.28 (October 21, 1999):

101 Added support for balanced transmission mode, creating the configuration macros `I1SCONFG_SUPPORT_BAL()`, `I1SCONFG_LINK_BALANCED()`, `I1SCONFG_LINK_DIR()`, `I1SCONFG_LINK_TOUT_ACK()` and `I1SCONFG_LINK_RETRIES()` to control this. The support code for balanced mode may be conditionally compiled, and excluded to save code and data space if not required.



Added macro `I1SCONFG_DIAG_STATISTIC_COUNTER` defining the storage size of diagnostic statistic values. This may be altered as required or when mapping these values to data objects. These counters are zeroed on reset, but do not rollover/wrap back to zero on overflow. If this macro is not defined, the diagnostic error statistic counters and support code are omitted, saving code and data space in devices that do not have diagnostic interface capabilities.



Renamed `I101DEFS_EI` to `I101DEFS_QDP_EI` and `I101DEFS_QUALITY_OV` to `I101DEFS_QDS_OV`



Renamed `FT12defs.h` to `IECFT12.h`, with corresponding changes to macro names. This is now shared with other TMW IEC products.



Continued revision of source code comments to enhance or more accurately document source code files.



Specific type casting added to prevent a number of compiler warnings.





For devices having more than one sector, corrected the evaluation of the period following a clock synchronization during which the library marks timestamps as valid. This period is set by the configuration parameter `I1SCONFG_CLOCK_VALID_PRD`, which was typically being divided by the number of sectors.












Fixed a problem that occurred when using 3-octet Information Object Addresses with a Motorola processor. The octet alignment was incorrect so that 3-octet IOA values were divided by 256 when written to the message stream. This fault has been present since version 1.06 (the initial release of the IEC 870-5-101 Slave Source Code Library).



Version 1.27 (June 30, 1999):

- 101 Created `I1SDATA_Cxxxx_FIND_POINT()` macros, and in `i1sdbas_storeCommandData()`, added calls to the new macros to help decrease execution-time for searching for a specific requested information object address (point number).
-  Now uses `TMWDEFS_COMPANY_*` strings instead of `I1MLOG_M101LIB_COMPANY_*` strings.
-  Continued revision of source code comments to enhance or more accurately document source code files.







Version 1.26 (June 22, 1999):

-  Created `TMWdtime.c` and `TMWdtime.h`. Replaced `IECDTIME` type with `TMWDTIME`, a more general date/time structure that is shared with other TMW products. More specifically, the `summertime` member was changed to `dstInEffect`, and the `year` member was changed to a `TMWDEFS_USHORT` that now ranges from 1901 to 2099 instead of 00 to 99.
-  Moved non-IEC-specific date/time functions from the `IECdtime` module to the `TMWdtime` module so that other TMW products can share them.
-  Throughout the IEC 870-5-101 Slave Source Code Library, changed the name of functions that included "TimeIEC" to "dateTime," and changed the name of variables that included "IECdTime" to "dateTime."
-  Added `I1SDATA_GET_SECTOR_DATE_TIME()` and `I1SDATA_SET_SECTOR_DATE_TIME()` macros, which, by default, call `I1STARG_GET_SYSTEM_DATE_TIME()` and `I1STARG_SET_SYSTEM_DATE_TIME()`, respectively. Changed parameters with which `I1STARG_GET_SYSTEM_DATE_TIME()` and `I1STARG_SET_SYSTEM_DATE_TIME()` are called. For sector time synchronization, changed to use `I1SDATA_SET_SECTOR_DATE_TIME()` instead of `I1STARG_SET_SYSTEM_DATE_TIME()`.
-  Changed `I1SANLYZ_FILTER_TRANS` to `I1SANLYZ_FILTER_TX` in order to be more consistent with other TMW products. Added `I1SANLYZ_FILTER_RX` to allow filtering only received messages.
-  Changed the order of parameters in `I1Sconfig.h` to be more consistent with config modules for other TMW Source Code Libraries.
-  Changed default value for `I1SCONFG_DIAG_STRINGS_ENABLED()` to `TMWDEFS_FALSE` because the most typical scenario is to save code-space by removing diagnostics and using a test Master station to perform protocol diagnostics.
-  Removed unnecessary `#include`'s to improve compile time.
-  Continued revision of source code comments to enhance or more accurately document source code files.




Version 1.25 (June 8, 1999):

-  Added `I1SDATA_RESET_POLL_TIMER()` macro, and call it from `i1sappl.c` whenever the device is class 1 or class 2 polled. This allows the database to time how often the device is being polled, and to take appropriate action. Since it is a macro, this functionality can be ignored without any code-space or execution time penalties.
-  Corrected a problem that prevented the sending of normal data change events when clock change events are disabled and time-synchronizations are received. This problem was introduced in Version 1.06.

Version 1.24 (May 20, 1999):

-  Added `I1SDATA_CLOSE()` macro, and pass a `causeOfInit` parameter to both `I1SDATA_CLOSE()` and `I1SDATA_INIT()` macros. `I1SDATA_INIT()` returns `TMWDEFS_BOOL` to indicate the success of initialization. These changes may be useful for implementations that allocate dynamic memory in `I1SDATA_INIT()`: Allocation errors should be reported by returning `TMWDEFS_FALSE`. Also, `I1SDATA_CLOSE()` should free any allocated memory.
-  Changed `i1starg.h` to use C-Macros instead of function prototypes. Converted `i1sdvrs_getMsTime()`, `i1sdvrs_getTimeIEC()`, and `i1sdvrs_setTimeIEC()` to `I1STARG_*` macros. Using C-Macros allows more efficient customization of the Source Code Library to fit unique requirements of the target application hardware. Also added `I1STARG_COMM_CLOSE()` to allow for communication ports to be closed upon remote resets (this, in turn, allows them to be re-opened in `I1STARG_COMM_OPEN()`).
-  Replaced `I1SCONFG_INTEL_BYTE_ORDERING()` and `i1sdbas_getXX()` and `i1sdbas_storeXX()` functions with `I1STARG_GETxx()` and `I1STARG_STORExx()` macros. Added `tmwintel` and `tmwmt1ra` modules.
-  Added `i1sdvrs_close101()` to call newly defined `I1SDATA_CLOSE()` and to call newly defined `I1STARG_COMM_CLOSE()` to execute a more orderly shutdown.
-  Created new `i1sanlyz_printf()` function and `I1SANLYZ_PUTS()` and `I1SANLYZ_ERROR()` macros to be a more versatile replacements for `i1sdvrs_analyzeText()` and `i1sdvrs_analyzeBuffer()`.
-  Changed name of version stamp variables (`I1SANLYZ_s101_*` to `I1SANLYZ_S101LIB_*`), and configuration parameters (`I1SCONFG_ANALYZER_SHOW_ON` to `I1SCONFG_DIAG_STRINGS_ENABLED`) to be more internally consistent.

Version 1.22 (April 6, 1999):

- 101 Changed the design of command responses by waiting for a response from the database (and possibly a down-stream device) before responding to the upstream device.
- 101 Improved support for DEACTCON with positive and negative confirmation of DEACTCON.
- 101 Changed **causeOfInit** to be preserved by the database for each sector rather than by the Source Code Library for each client (remote Master). This is necessary because the database may have knowledge of remote initialization of downstream sectors.
- 101 Now allows configurable support for different qualifiers of the reset process command.
- 101 No longer requires command points to have corresponding monitored points.
- 101 Moved **I1SCONFG_SIZE_CAUSE_OF_TRANS()** from **i1sconfig.h** to **I1SAPPL_SIZE_CAUSE_OF_TRANS()** inside **i1sappl.c** because only a value of 1 is fully supported. Therefore, currently, it is not a configurable parameter.
- 101 Added client index, sector index, and point number to **i1sdvrs_setTimeIEC()** to allow time to be transferred, by the database, through sectors to downstream devices.
-  Now uses **tmwprntf_s()** instead of **sprintf()**. **tmwprntf.c** and other **tmwxxxxx** modules are now provided as part of the Source Code Library. These modules provide utility support for embedded compilers that may not include such support.
-  Added **I1SANLYZ_S101_*** literal constants to **i1sanlyz.h** so that the product name and version can be used by external application-specific software.
-  Removed **_CAN_STORE()** and **_SELECT_REQUIRED()** macros; combined their functionality into **_STORE()** macros. Added **I1SDBAS_COMMODE** and cause of transmission parameters to **_STORE()** macros. Renamed **_FINISHED()** macros to **_STATUS()** macros. All **_STORE()** and **_STATUS()** macros now return the **I1SDBAS_COMSTAT** enumerated type instead of a Boolean. Added new types: **I1SDBAS_COMSTAT**, **I1SDBAS_COMMODE**, **I1SDBAS_FEEDBACK_FUNCTION**, and **I1SDBAS_STATUS_FUNCTION**. These types are used by **i1sdbas_storeCommandData()**, and allow the removal of **i1sdbas_commandFinished()**, and **i1sdbas_commandFeedback()**.



Changed the names of many configuration parameters to be more consistent with their usage.

Old Name	New Name
I1SCONFG_ANALYZER_ID_xxxx	removed (was not used)
I1SCONFG_SIZEOF_LARGEST_COMMAND	I1SAPPL_SIZEOF_LARGEST_COMMAND
I1CONFG_SIZE_CAUSE_OF_TRANS	I1SAPPL_SIZE_CAUSE_OF_TRANS
I1SCONFG_INTER_CHAR_TIMEOUT	I1SCONFG_PHYS_CHAR_TIMEOUT
I1SCONFG_TOTAL_FRAME_TIMEOUT	I1SCONFG_PHYS_FRAME_TIMEOUT
I1SCONFG_LINK_ADDRESS_SIZE	I1SCONFG_LINK_SIZE_ADDRESS
I1SCONFG_SIZE_ADDRESS_ASDU	I1SCONFG_ASDU_SIZE_CMN_ADDR
I1SCONFG_SIZE_POINT_NUM	I1SCONFG_ASDU_SIZE_IOA
I1SCONFG_CYCLIC_REPORTING_INTERVAL	I1SCONFG_APPL_CYCLIC_PRD
I1SCONFG_SELECT_ARM_TIMEOUT	I1SCONFG_APPL_SELECT_TIMEOUT
I1SCONFG_C_SE_ACTTERM_USED	I1SCONFG_APPL_USE_C_SE_ACTTERM
I1SCONFG_NUM_SECTORS	I1SCONFG_DBAS_NUM_SECTORS
I1SCONFG_SECTOR_ADDRESS	I1SCONFG_DBAS_SECTOR_ADDRESS
I1SCONFG_RBE_SCAN_PERIOD	I1SCONFG_DATA_SCAN_PRD
I1SCONFG_Mxxxx_SCAN_FOR_CHANGES	I1SCONFG_Mxxxx_SCAN_ENABLED
I1SCONFG_Mxxxx_ONLY_MOST_RECENT	I1SCONFG_Mxxxx_ONLY_RECENT
I1SCONFG_INIT_PROPAGATION_DELAY	I1SCONFG_CLOCK_INIT_PROP_DLY
I1SCONFG_CLOCK_VALID_INTERVAL	I1SCONFG_CLOCK_VALID_PRD
I1SCONFG_COMMAND_CLASS_ID	I1SCONFG_COMMANDS_CLASS_ID
I1SCONFG_PARAMETER_CLASS_ID	I1SCONFG_PARAMETERS_CLASS_ID
I1SCONFG_CLOCK_SYNC_IOA	I1SCONFG_OPT_CLOCK_SYNC_IOA
I1SCONFG_COMMAND_RETURN_INFO	I1SCONFG_OPT_USE_C_ACTTERM
I1SCONFG_FREEZE_RETURNS_COUNTERS	I1SCONFG_OPT_FRZ_RTN_DATA
I1SCONFG_SINGLE_CHAR_ACK	I1SCONFG_OPT_ONE_CHAR_ACK
I1SCONFG_SINGLE_CHAR_RESPOND	I1SCONFG_OPT_ONE_CHAR_RSP
I1SCONFG_QTY_IN_READ_RESPONSE	I1SCONFG_OPT_READ_RTN_QTY
I1SCONFG_ANALYZER_SHOW_ON	I1SCONFG_DIAG_STRINGS_ENABLED
I1SCONFG_ANALYZER_ID_SHOW	I1SCONFG_DIAG_ID_AUX_MASK



Corrected a parameter naming problem within the `I1STASK_FIXED_TOTAL_LENGTH()` macro.






Corrected a potential problem inside `dbasReadGroupPmena()` and `dbasReadGroupPmenb()`. These functions now initialize `qualifier=0`. Without this initialization, the qualifier returned as part of a parameter read may have been random (un-initialized data), depending up on the implementation of `I1SDATA_PMENA_READ()` and `I1SDATA_PMENB_READ()`. The example implementations in I1Ssim did not cause such an error.



Because we no longer require command points to have corresponding monitored points, we must now check for a valid monitored point in the `i1sdbas_cxxxxFeedback()` functions. In addition to the added functionality this provides, this also corrects a problem that had existed in `i1sdbas_commandFeedback()`. The problem was that the return value from `I1SDATAT_Cxxxx_GET_TABLE_ENTRY()` was being used to test for a valid monitored point. This was incorrect because some database implementations may not be table-driven and will always return `TMWDEFS_NULL` from the `I1SDATA_Cxxxx_GET_TABLE_ENTRY()` macros. Now, we test the value returned by `I1SDATA_Cxxxx_Mxxxx_POINT_NUM()`.


Version 1.20 (July 7, 1998):

-  Corrected a problem with sending corrupted clock change events in 56-bit time format.
-  Corrected potential problem with spontaneous response messages exceeding buffer size because `buildSpontaneousResponse()` used OR condition instead of AND condition when checking buffer size and number of points.
-  Corrected problem with quantity of points in read function responses. The quantity was 1 too many because compared `maxQuantity` using less than or equals in `dbasReadGroup()` functions.

Version 1.18 (June 8, 1998):

-  Corrected a problem with 3-byte information object addresses in functions `ilsappl_getPointNum()` and `ilsappl_storePointNum()`.

Version 1.16 (June 3, 1998):

- 101 Added load transmission propagation delay (ASDU type 106 in the control direction). This value is then used in the clock synchronization procedure: The transmission propagation delay is added to the time received in the clock synchronization message before it is stored in the target system clock. The delay is also subtracted from the target system time placed in the clock synchronization ACTCON message.
-  Corrected syntax variations that caused errors with some embedded compilers.

Version 1.14 (June 2, 1998):



- 101 Added support for freezing and resetting of counters (ASDU type 101).
- 101 Added support for loading parameters (ASDU types 110 and 111).

Version 1.12 (May 29, 1998):




- 101 Added support for the reset process command (ASDU type 105).
- 101 Enhanced support for the read command (ASDU type 102) and added support for counter interrogations (ASDU type 101).
- 101 Added support for primary link function codes 1 and 8 (reset of user process and request for access demand).

- 101 Separated **I1SDATA** command macros from monitored data macros so that they can be configured, by a specific database implementation, to use different information object addresses and different access methods. Added macros for determining when commands are “finished,” and for determining if monitored data has changed (Now, the definition of what constitutes a “change” in a data value is under the total control of the database manager, and is outside the scope of the Source Code Library).
- 101 Added support for normalized measurands (ASDU types 9, 10, 48, and 110).
- 101 Now supports sending spontaneous data to multiple 101 masters.
- 101 Added support for extended time format (CP56time2a). Spontaneous data can be sent with normal 24-bit time format, 56-bit time format, or with no time information at all. The reporting of spontaneous clock data (hour rollover) can now be configured by specifying the class (1, 2, or disabled).
- 101 Added configuration for the class (1, 2, or disabled) for spontaneous ASDU data types where either the specification is ambiguous or where some applications conflict with the specification.
- 101 Added spontaneous data support for double points (ASDU types 3, 4, or 31), regulating steps (ASDU types 5, 6, or 32), normalized measurands (ASDU types 9, 10, or 34), scaled measurands (ASDU types 11, 12, or 35), and integrating totals (ASDU types 15, 16, or 37)

Version 1.10 (April 19, 1998):

- 101 Added support for the read command (ASDU type 102).
-  Made the size of the link address a configuration parameter and put it in `i1sconfg.h` for either compile-time or run-time variability.
-  Added a configuration parameter to specify the use of single character ACKs.

Version 1.08 (April 17, 1998):

- 101 Added support for ASDU types `M_ST_NA` (5), `M_IT_NA` (15), `C_DC_NA` (46), `C_RC_NA` (47), and `P_ME_NB` (111).
-  Converted to new corporate coding standard.
-  Created report-by-exception module to scan static database for changes, buffer them, and supply them in IEC 870-5-101 format to the application layer when spontaneous data is requested. Added configuration parameters for this module.
-  Combined `respondClass1()` and `respondClass2()` functions into a single `respondClassx()` function to allow better control of priority when data can be sent in either frame type.



Added `i1stask_checkInvalidTime()` to set invalid status if clock has not been synchronized within a configurable period of time.



Add function `iecdtime_store3InMessage()` to store time in message in shortened 3 octet format.

Version 1.06 (April 2, 1998):

Initial release of the IEC 870-5-101 Slave Source Code Library.