

# Release Notes for USB2ANY.DLL

---

---

## **v2.8.2.0**

### **General Release**

1. Fixed a bug that prevented an application program from getting a plug/unplug callback notification if the application was not processing received Windows messages.

## **v2.8.0.0**

### **General Release**

## **v2.7.0.17**

2. Suppress SMBus error if bytes received is less than requested and PEC is on.
3. Added u2aMSP430\_MemoryWrite and u2aMSP430\_MemoryRead API functions.
4. Added support for SPI streaming (currently disabled).
5. Added SPI\_WriteAndReadEx function that allows CS to be specified.
6. Added Port\_WritePulse function.
7. Added mode to keep MOSI high between transactions (SPI interface).
8. Added API function u2aStream\_Execute().
9. Added API function u2aPort\_WritePulse().
10. Added new C# functions to USB2ANY\_CS.DLL:  
SendRawDataBuf, SendCommand, GetCommandResponse, SMBUS\_Control,  
SMBUS\_SendByte, SMBUS\_WriteByte, SMBUS\_WriteWord, SMBUS\_WriteBlock,  
SMBUS\_ReceiveByte, SMBUS\_ReadByte, SMBUS\_ReadWord, SMBUS\_ReadBlock,  
SMBUS\_BlockWriteBlockRead.
11. Added API functions: u2aSendRawDataBuf, u2aSendCommand, and u2aGetCommandResponse.
12. Fixed bugs in u2aSMBUS\_WriteBlock() function.
13. Fixed bugs in u2aSMBUS\_ReadBlock() function.

## **v2.7.0.7**

1. Enable the SPI\_WriteAndReadEx() function, which is essentially the same as the SPI\_WriteAndRead() function, except that it has an additional parameter that allows the user to specify a GPIO to be used as the SS signal.

## **v2.7.0.0**

## **v2.6.4.0**

---

### **Bug fixes, new features, etc.:**

1. Added ability to use the `MOST_RECENT_HANDLE` constant for the handle parameter, instead of an actual handle
2. Added asynchronous I/O handlers
3. Added asynchronous I/O option to `u2aI2C_InternalRead()`
4. Added code to log data read by `u2aSPI_WriteAndRead`
5. Added code to log progress of firmware update
6. Added code to prevent problems if there is no parent dialog for the firmware loader
7. Added code to remove extra junk (left by CCS) at the end of the hex file
8. Now logs a debug comment "Using `USB2ANY_CS.DLL`" when `u2aOpen` is called from `USB2ANY_CS.DLL`
9. Added error code `ERR_COMPARE_FAILED` and `ERR_FILE_TOO_BIG`
10. Added error codes `ERR_INVALID_PARAMETER` and `ERR_NOT_INITIALIZED`
11. Added FIFOs for `I2CRead`, `I2CBlkWriteBlkRead`, `I2CReadWithAddress`, and `I2CReadInternal`
12. Added `GPIO_WritePulse()` function
13. Added `MOST_RECENT_HANDLE` constant
14. Added streaming option to `u2aI2C_BlockWriteBlockRead()`
15. Added streaming option to `u2aI2C_InternalRead()`
16. Added streaming option to `u2aI2C_MultiRegisterRead()`
17. Added streaming option to `u2aI2C_RawRead()`
18. Added streaming option to `u2aI2C_RegisterRead()`
19. Added `u2aDisplayScale_Read()` function (currently disabled)
20. Added `u2aDisplayScale_ReadReg()` function
21. Added `u2aDisplayScale_Setup()` function
22. Added `u2aDisplayScale_Write()` function (currently disabled)
23. Added `u2aDisplayScale_WriteAndRead()` function
24. Added `u2aDisplayScale_WriteReg()` function
25. Added `u2aGPIO_WritePulse()` function
26. Added `u2aPegasus_Test()` function
27. Added `u2aSetAsyncIOCallback()` function
28. Added `u2aStream_Next()`, `u2aStream_Stop()`, `u2aStream_Status()`, and `u2aStream_GetData()` functions
29. Added `u2aSuppressPopups()` API function (mainly for Web apps)
30. Adjusted some delays for improved efficiency and stability
31. Improved error checking and reporting
32. Improved initialization of Debug Logging, Packet Logging, and API Profiling
33. Improved logging and reporting of status
34. Make sure debug settings are initialized before changing or reporting debug logging state.
35. Modified most C# functions (in `USB2ANY_CS.dll`) to allow an optional handle parameter

## **v2.6.3.10 (beta)**

---

### **Bug fixes, new features, etc.:**

1. When converting firmware hex file, the code is now split into three parts (instead of two) to prevent static memory objects from overflowing.
2. The firmware converter now looks for the firmware file in the `USB2ANY_Workspace_V2.7` folder.

3. The firmware converter now checks the generated source file for limit violations.
4. Added a CheckResult function to ExampleBase.cpp to allow example programs to easily check for and report errors.
5. Added a button to display debug settings in the USB2ANY Support Info dialog.
6. Modified code to allow a silent (i.e., no messages, popups, etc.) firmware update.
7. The handle of the most recently opened connection is now saved (until it is closed). It can be used by passing the MOST\_RECENT\_HANDLE macro constant as a handle to any API function. This eliminates the need to save the handle in applications where that is not possible (e.g., Java plugins).
8. Added the u2aSuppressPopups() function to allow the firmware update dialog from being displayed.

## **v2.6.3.0**

### **Bug fixes, new features, etc.:**

1. Debug settings for Debug Logging, Packet Logging, and Profiling are saved/restored to/from a settings file specific to the application program that loaded USB2ANY.DLL. This allows the settings to be remembered from session to session.
2. The current default Debug Logging, Packet Logging, and Profiling settings are now logged to the debug log at the beginning of each session.
3. Added code to display a message box, with a performance warning, at the beginning of a session if Packet Logging and/or Profiling are enabled.
4. The full path of the program that loaded USB2ANY.DLL is now logged to the debug log at the beginning of each session.
5. Added new API function, u2aUART\_SetMode(), to allow the setting of special UART modes.
6. Added UART\_Mode enums UART\_Normal, UART\_ReceiverOff, and UART\_RecvAfterXmit.
7. Some improvements to Packet Logging.
8. Some improvements to Debug Logging of UART functions.

## **v2.6.2.25 (beta)**

### **Bug fixes, new features, etc.:**

1. Added source file USB2ANY\_Firmware.cpp (generated by ConvertUSB2ANYHexFile.exe).
2. Many improvements and bug fixes to the firmware loader.
3. USB2ANY.DLL now requires a matching version of firmware.
4. The required firmware version is now embedded into USB2ANY.DLL.
5. Loader dialog now displays "Please wait..." while waiting for BSL mode.
6. USB2ANY.DLL no longer depends on BSL.DLL (the required code is now built-in).

### **Other changes:**

7. Added GetRequiredFirmwarePath().
8. In ParseFirmwareVersion(), dwVersion is updated only if it is not NULL.
9. Minor improvements to RunProcess().
10. Added CalcCRC().

## v2.6.2.20 (beta)

### **Bug fixes, new features, etc.:**

1. Added source file USB2ANY\_Firmware.cpp (generated by ConvertUSB2ANYHexFile.exe)
2. Added source files BSLButtonHelp.h and BSLButtonHelp.cpp to display BSL help.
3. Improved code used to locate the SDK folder. It now works correctly when the SDK is installed into a non-standard location.
4. Added CU2AClass::EasyScale\_WriteAndRead() function.
5. Added CU2AClass::OneWire\_PulseWriteEx() function.
6. Marked OneWire\_Channel enums as deprecated.
7. Deprecated OneWire\_Mode enum OW\_EnablePulseMode.
8. Added OneWire\_Mode enums OW\_Mode1, OW\_Mode2, and OW\_Mode3.
9. Added OneWire\_Flags enums OW\_FLAG\_NONE, OW\_FLAG\_INVERT, and OW\_FLAG\_NO\_INIT.
10. Added u2aSMBUS\_BlockWriteBlockRead() function.
11. Added u2aI2C\_BlockWriteBlockRead(0) function.
12. Added u2aEasyScale\_WriteAndRead() function.
13. Added u2aOneWire\_PulseWriteEx() function.
14. Added u2aUART\_GetRxCount() function.
15. Increased range of *milliseconds* parameter for SetReceiveTimeout() function from 2 - 100 to 2 - 200 .
16. Added U2A\_ApiClassV2::I2C\_BlockWriteBlockRead() function.
17. Added U2A\_ApiClassV2::UART\_GetRxCount() function.
18. Added U2A\_ApiClassV2::EasyScale\_WriteAndRead() function.
19. Added U2A\_ApiClassV2::OneWire\_PulseWrite() function.

### **Other changes:**

20. Added code to prepare for static firmware loader.
21. Added Cmd\_EasyScale\_WriteAndRead.
22. Modified to use version string from Version.h (several places).
23. Added source files ReceiveTimeoutDialog.h and ReceiveTimeoutDialog.cpp to display a dialog that allows the user to change the receive timeout.

## v2.6.2.9 (beta)

### **Bug fixes, new features, etc.:**

1. Added API function u2aOneWire\_PulseSetup
2. Added API function u2aOneWire\_PulseWrite
3. Added API function u2aOneWire\_SetMode
4. Added API function u2aOneWire\_SetState
5. Added ERR\_NOT\_ENABLED error code
6. Added example code for OneWire interface
7. Added OneWire tab to USB2ANY Explorer
8. Fixed bug that caused the wrong baud rate to be logged
9. Fixed some baud rate selection issues
10. Added "BSL Help" button to Firmware Update dialog

### Other changes:

11. Added CU2AClass::OneWire\_PulseSetup()
12. Added CU2AClass::OneWire\_PulseWrite()
13. Added CU2AClass::OneWire\_SetMode()
14. Added CU2AClass::OneWire\_SetOutput()
15. Added Cmd\_I2C\_BlkJWriteBlkJRead
16. Added Cmd\_OneWire\_PulseSetup
17. Added Cmd\_OneWire\_PulseWrite
18. Added Cmd\_OneWire\_SetMode
19. Added Cmd\_OneWire\_SetState
20. Added OneWire\_Mode and OneWire\_Channel typedef enums
21. Changed Cmd\_I2C\_Read\_Internal to Cmd\_I2C\_ReadInternal
22. Changed Cmd\_I2C\_Write\_Internal to Cmd\_I2C\_WriteInternal
23. Added new version string macros
24. Added code to UART\_Read to attempt read multiple times
25. Added U2A\_ApiClassV2::OneWire\_PulseSetup()
26. Added U2A\_ApiClassV2::OneWire\_SetMode()
27. Added U2A\_ApiClassV2::OneWire\_SetOutput()
28. Changed Cmd\_DigitalCapture to Cmd\_Digital\_Capture
29. Added PinManager code for OneWire
30. Added support for OneWire interface
31. Added u2aOneWire\_PulseSetup()
32. Added u2aOneWire\_SetMode()
33. Added u2aOneWire\_SetOutput()
34. Added code to display the BSLButtonLocator dialog when the help button is clicked

## **v2.6.2.0**

1. Added u2aSuppressFirmwareCheck() function
2. Changed splash screen code to not display splash if main window was not found (caused crash on XP).
3. Fixed some logic bugs in the firmware update code.
4. Added code to calculate and check PEC codes for SMBus functions.
5. Fixed endian bug in u2aSMBUS\_ReadWord() function.
6. Added handle checking code to u2aFEC\_PurgeBuffer() and u2aFEC\_GetResult() to prevent crash if USB2ANY is not open.
7. Several places, changed value of invalid handle from '0' to 'INVALID\_U2A\_HANDLE'
8. Several places, changed method of validating handle from 'h == 0' to 'IsValidU2A\_HANDLE(h)'
9. Added code to validate handle in GetApiPointer() function
10. Added code to validate handle in u2aStatus\_GetErrorCode() function
11. Fixed bug in u2aStatus\_IsUSB2ANYConnected
12. Fixed bug in u2aGetErrorList
13. Fixed bug in u2aStatus\_GetErrorCode
14. Improved error checking in u2aSMBUS\_Control

15. Fixed return value bug in u2aGPIO\_ReadState
16. Added u2aUART\_DisableReceiver function
17. Improved error checking in u2aADC\_ConvertAndRead
18. Fixed return value bug in u2aADC\_ConvertAndRead
19. Fixed return value bug in u2aMSP430\_WordRead
20. Fixed incorrect return value in u2aMSP430\_ByteRead
21. Improved error checking in u2aFEC\_PurgeBuffer
22. Improved error checking in u2aFEC\_GetResult
23. Fixed bugs and improved error checking in u2aStatus\_EVMDetect
24. Added u2aGetDebugLogPath function
25. Added u2aDigital\_Capture, u2aDigital\_GetData, and u2aDigital\_GetStatus functions
26. Changed logic in u2aEnableDeviceDetect()
27. Added u2aSetDeviceDetectVidPid()
28. Changed logic in u2aEnableDeviceDetectMsg()
29. Removed check for (s\_MsgWnd == NULL) in u2aStatus\_IsUSB2ANYConnected()
30. Added code in u2aOpen() to handle s\_bSuppressFirmwareCheck flag
31. In u2aSMBUS\_ReceiveByte(), don't check PEC if I2C read failed
32. In u2aSMBUS\_ReceiveByte(), use CalcCRC8() instead of CalcPEC to check PEC
33. In u2aSMBUS\_ReadWord(), use CalcCRC8() instead of CalcPEC to check PEC
34. In u2aSMBUS\_ReadBlock(), fixed bugs in PEC calculation
35. Added code to call u2aInvokeBSL when Firmware Loader dialog opens.
36. Changed SetReceiveTimeout() to return ERR\_PARAM\_OUT\_OF\_RANGE if the *milliseconds* parameter is less than 2 or greater than 100
37. Added code to handle Digital Capture functions
38. Modified LED\_SetState to send the BlinkCode parameter
39. Added commands for Digital Capture
40. FirmwareLoader: Fixed bug that caused crash on second programming attempt
41. FirmwareLoader: Updated BSL code to version 00.07.88.38
42. FirmwareLoader: Increased delay before InitializeBSL(), to prevent "USB device was unplugged" error
43. FirmwareLoader: Added code option to have OnDeviceChange() ignore VID/PID
44. FirmwareLoader: Added CHiddenMsgWindow::SetDeviceID()
45. The firmware loader will now display "Firmware version 1.x.x.x is deprecated..." message before loading firmware older than v2.x.x.x.

## v2.6.1.20 (beta)

1. Added macros INVALID\_U2A\_HANDLE and IsValidU2A\_HANDLE() to help validate handles
2. Added functions: CU2AClassUART\_DisableReceiver(), Digital\_Capture(), Digital\_GetData, and Digital\_GetStatus().
3. Added LEDS enum for LED states.

## v2.6.1.0

1. Requires firmware version 2.6.1.0, or later.
2. Added u2aSuppressFirmwareCheck() function.
3. Changed splash screen code to not display splash if main window was not found (caused crash on XP).
4. Fixed some logic bugs in the firmware update code.
5. Added code to calculate and check PEC codes for SMBus functions.
6. Fixed endian bug in u2aSMBUS\_ReadWord() function.
7. Added error code ERR\_DATA\_CRC\_FAILED.

## v2.6.0.5 (beta)

1. Support for firmware version 2.6.0.5.
2. Change ERR\_NO\_EVM text from "No EVM" to "EVM is required for external power".
3. Added RFFE commands:

Cmd\_RFFE\_RegZeroWrite

Cmd\_RFFE\_RegWrite

Cmd\_RFFE\_ExtRegWrite

Cmd\_RFFE\_ExtRegWriteLong

Cmd\_RFFE\_RegRead

Cmd\_RFFE\_ExtRegRead

Cmd\_RFFE\_ExtRegReadLong

4. SPI: To reduce confusion, the SPI\_ClockPhase enums have been changed:

From:

SPI\_Change\_On\_First\_Edge = 0

SPI\_Change\_On\_Following\_Edge = 1

To:

SPI\_Capture\_On\_Trailing\_Edge = 0

SPI\_Capture\_On\_Leading\_Edge = 1

The original enums are still supported, but have been deprecated.

5. SPI: To reduce confusion, the SPI\_LatchPolarity type name has been changed to the new SPI\_ChipSelectPolarity type.
6. SPI\_LatchPolarity has been defined as a synonym for SPI\_ChipSelectPolarity.
7. SPI: To reduce confusion, the SPI\_ChipSelectPolarity enums have been changed:

From:

SPI\_Low\_To\_High

SPI\_High\_To\_Low

To:

SPI\_CS\_Active\_Low

SPI\_CS\_Active\_High

The original enums are still supported, but have been deprecated.

8. Added SMBUS\_PEC\_CRC8 as a synonym for SMBUS\_PEC\_ON.
9. Added the following new API functions:
  - u2aEnablePacketLogging()
  - u2aSuppressSplash()
  - u2aLogComment()

- u2aDAC\_SetVoltage()
  - u2aFEC\_GetResult()
  - u2aFEC\_PurgeBuffer()
  - u2aPower\_ReadStatus()
10. Deprecated the following API functions:
    - GetSerialNumberFromHandle() (replaced by u2a GetSerialNumberFromHandle)
    - u2aReadResponse() (obsolete)
  11. Fixed a bug in I2C\_MultiRegisterWrite() that caused addressing problems.
  12. Added a new Firmware Loader dialog that shows all available devices that can be updated and allows the user to select which one to update.
  13. Added an “Update to firmware version” list box to the Firmware Loader dialog, which lists the available versions of firmware that can be loaded. The recommended version is automatically pre-selected, but the user can change the selection to an alternate version.
  14. Fixed a bug that prevented multiple boards from being opened (and used) from a single instance of the USB2ANY.DLL library. The DLL now supports up to six simultaneous connections.
  15. By default, USB2ANY.DLL displays a splash screen when a program attaches to it. The splash screen can be suppressed by calling the new u2aSuppressSplash() function with a TRUE parameter before calling other API functions.
  16. Fixed a bug in the u2aFirmwareVersion\_Read() function that erroneously caused it to return ERR\_OPERATION\_FAILED if the bufsize parameter was set less than 5. It now correctly returns ERR\_BUFFER\_TOO\_SMALL if the bufsize parameter is set less than 4.

**NOTE: The following changes are internal and not directly visible to end users.**

17. Improvements to ByteFifo class.
18. Added line and function parameters to the ERROR\_EXIT macro.
19. Changed GetLocalAppDataPath(), WriteSetting(), and ReadSetting() functions to use CSIDL\_LOCAL\_APPDATA instead of CSIDL\_COMMON\_APPDATA for data storage.
20. Added GetSDKPath() function;
21. Added MakeFirmwarePath() function.
22. Changed FindNewestFirmware() function to create a list of all available firmware versions, in addition to finding the newest version.
23. Added GetAvailableFirmwareVersions() functions to get the list of all available firmware versions.
24. Added a RunProcess() function to execute a child process.
25. Completely changed (and simplified) the code used to display the splash screen. The old code had several problems, including lock-ups and crashes.
26. USB2ANY\_SDK.H: Fixed comments that showed the wrong ID numbers for several commands.
27. Added Cmd\_InvokeBSL and Cmd\_FirmwareDebugMode commands.
28. Added a BOOL bEnumOnly parameter to the u2aFindControllersBSL() function.
29. Changed communications code to use a CByteFifo for ADC results.
30. The enabler for packet logging is now persistent (previous logging state is remembered).
31. Command names are now shown in the packet log, instead of hex codes.
32. Received packets are now synchronized using semaphore objects.
33. Asynchronous “Payload” packet types are parsed in a CriticalSection.



34. TI\_USB2ANY\_Init() function call was removed from the U2A\_ApiClass constructor. This was causing open handles to be lost.
35. Individual semaphore objects are now used for each interrupt “channel”, instead of just a single semaphore for all channels.
36. Added an ADC\_Read() function to get data received asynchronously.
37. Added code to ADC\_Acquire() and ADC\_AcquireTriggered() to flush the FIFO before sending the command to the firmware.
38. Changed ADC\_GetStatus() to get buffered status that was received asynchronously instead of requesting status from the firmware.
39. Added ADC interrupt handler.
40. Changed the EXIT\_FUNCTION macro to return a value.
41. Added EXIT\_FUNCTION\_PTR macro.
42. Added internal FirmwareLoaderDialog() function.
43. Added BOOL bEnumOnly parameter to the u2aFindControllersBSL() to allow it to be used to enumerate controllers in BSL mode, without displaying the Firmware Loader dialog.
44. Added a call to TI\_USB2ANY\_Init() to the u2aOpen() function. This will delay the display of the splash screen until an attempt is made to actually open a connection to a board.
45. Added code to u2aOpen() that prevents checking of the firmware version if g\_nFirmwareDebugMode is set to FW\_DEBUG\_NO\_CHECK.
46. Added code to display the board’s serial number in the “firmware version ... is not supported” message box.
47. Added code to display the board’s serial number in the “newer version of firmware is available” message box.
48. All SMBus commands are now emulated by the DLL using I2C commands:
  - u2aSMBUS\_SendByte()
  - u2aSMBUS\_WriteByte()
  - u2aSMBUS\_WriteWord()
  - u2aSMBUS\_WriteBlock()
  - u2aSMBUS\_ReceiveByte()
  - u2aSMBUS\_ReadByte()
  - u2aSMBUS\_ReadWord()
  - u2aSMBUS\_ReadBlock()
  - u2aSMBUS\_Control()
  - u2aSMBUS\_GetEchoBuffer()
49. Fixed a bug that caused the u2aI2C\_MultiRegisterRead() function to return one extra byte.
50. Fixed a bug in u2aFEC\_Configure() that could prevent it from returning ERR\_PARAM\_OUT\_OF\_RANGE when an invalid parameters was passed to it.

**KNOWN ISSUES:**

51. The following functions are not implemented:
  - u2aMSP430\_MemoryRead()
  - u2aMSP430\_MemoryWrite()

## **v2.5.1**

---

1. Support for firmware version 2.5.1.0.

## **v2.5.0**

---

1. USB2ANY.dll now displays a splash screen on startup. BETA versions are labeled as such.
2. Improved error handling.
3. More detailed debug logging.
4. All API functions now return error codes returned by the firmware.
5. u2aFindControllers can now locate USB2ANY boards that are executing BSL code.
6. When u2aOpen() is called, it now checks the current firmware version of the attached controller and compares it with the newest known firmware version. If the firmware can be updated, the user is prompted to do so. If the current firmware version is less than the minimum required version, the update is mandatory (otherwise, the open will fail).
7. Fixed a bug in I2C\_MultiRegisterRead() that caused it to return invalid data.
8. Fixed a bug in I2C\_ReadInternal () that caused it to return invalid data and/or byte count.
9. Added additional baud rates to u2aUART\_Control() function: 300, 320, 600, 1200, 2400, and 4800.
10. Added u2aADC\_AcquireTriggered() function.
11. Removed all references to FEC1, FEC2, and FEC3, since the board has only one FEC port (FEC0).
12. Fixed some bugs in the FEC code and verified that it is working.
13. Added high-resolution profiling code to all API functions.
14. Added u2aEnableAPIProfiling() function to enable/disable API profiling.
15. Added u2aProfileTimestamp() function to allow a high-resolution timestamp to be inserted into the debug log at any time.
16. Added u2aPower\_Notify() function to enable/disable an asynchronous callback function that will be called whenever a power event occurs.
17. Fixed bugs in u2aEasyScale\_Read() function that caused incorrect data to be returned.
18. Changed name of u2aDAC\_SetVoltage() API function to u2aDAC\_SetValue(), to better match its purpose.
19. Changed name of u2aPower\_SetVoltage() API function to u2aPower\_SetVoltageRef(), to better match its purpose.

***NOTE: The following changes (in blue) are internal and not directly visible to end users.***

20. Added optional debug logging to HID driver.
21. Changed maximum size limit for packet logs to 50 Kbytes.
22. Logging to SysDebug now defaults to disabled.
23. Packet logging now defaults to enabled.
24. The single packet receive queue has been replaced by packet queues for each distinct command.
25. FEC responses are no longer queued, but are stored in a global structure.
26. Received data from u2aUART\_Read operations is now stored in a 1024-byte FIFO buffer.
27. Packet log files are now prevented from growing uncontrolled. On initialization, the packet log is archived, and a new one started, if the file size is greater than 50K bytes.

28. New packet log sessions are now timestamped.
29. The format of packet log entries has been changed and now includes the sequence number, packet type, command, packet length, PEC, status code, payload length, and payload data (if applicable).
30. A callback handler has been added for power events.
31. Improved logging of packet read errors.
32. Added code to detect and report orphaned packets.
33. Added code to detect and report payload size errors.
34. Added handler for payload packets.
35. Cleaned-up SendBuffer function.
36. Improved error checking in SendCommandPacket function.
37. Added FEC response handler.
38. FindDevices function can now optionally ignore serial numbers. This is user to detect BSL firmware.
39. Many commands have been update to use the “new” protocol for communication between USB2ANY.DLL and the USB2ANY firmware.
40. Reliability improvements to ReadResponse function.
41. DACs\_Write command now allows access to DAC2 (only for OneDemo).
42. DACs\_Write now differentiates the DAC type (DAC5574 for OneDemo, DAC5571 for USB2ANY).
43. Added ADC\_AcquireTriggered function for delayed triggering of the ADC.
44. Implemented EasyScale\_Control, EasyScale\_Write, and EasyScale\_Read commands.
45. Implemented Power\_Notify command to allow asynchronous power change notifications (via callback).

## **v2.1.0**

1. API now supports plugging/unplugging of a USB2ANY.
2. Improvements to **u2aEnableDeviceDetect** API function.
3. Added **u2aIsUSB2ANYConnected** function to allow connection status to be obtained.
4. Fixed a bug that caused the **u2aPower\_ReadStatus** function to fail.
5. Fixed a bug that caused the **u2aPower\_ReadStatus** function to fail when in Polled Mode.
6. Fixed a bug that caused the **u2aI2C\_RegisterRead** function to return wrong results.
7. Fixed a bug that caused the **u2aI2C\_MultiRegisterRead** function to return wrong results.
8. Fixed a bug that caused the **u2aI2C\_RawRead** function to return wrong results.
9. Modified **u2aGPIO\_ReadPort** API function to return correctly when in Polled Mode.
10. Modified **u2aSPI\_WriteAndRead\_P** API function to return correctly when in Polled Mode.
11. Modified **u2aUART\_Read** API function to return correctly when in Polled Mode.
12. Modified **u2aPower\_ReadStatus** API function to return correctly when in Polled Mode.
13. Many improvements to error detection and logging.
14. Added **u2aEnableDebugLogging** API function.
15. Improvements to reliability and functionality of **u2aGetErrorList** API function.
16. Added **u2aStatus\_GetErrorCode** API function, which returns a code for the first error that occurred during execution of the last called API function.
17. Added **u2aStatus\_GetText** and **u2aStatus\_GetTextW** (Unicode version) functions to get a string containing the text for an error code.

18. The **u2aGetStatusText** API function is now deprecated. Use **u2aStatus\_GetText** instead.
19. A bug was fixed that prevent a debug log from being created if the **USB2ANY\Logs** sub-folder did not exist in the default Windows Documents folder. Now, the required sub-folder(s) will be created, if they do not already exist.
20. Added **u2aGetSerialNumberW** function to get a Unicode string containing the serial number of the specified board.
21. Added **u2aOpenW** function to open a controller using a Unicode string containing the serial number.
22. Various improvements to the **u2aReadResponse** API function.
23. By default, a new debug log file will be created when the SDK DLL is loaded. If a previous log file exists, with the same name, it will be saved with **.bak** appended to the name, overwriting any previous **“.bak”** file.
24. If append mode is enabled for debug log files, the default maximum file size is 1Mb. If the file grows beyond that limit, it will be saved and a new log file started.
25. A bug was fixed that could cause several extraneous log files (one for each running application that attached the DLL) to be opened when the SDK DLL was loaded in a debugger.

**NOTE: The following changes (in blue) are internal and not directly visible to end users.**

26. Fixed a bug that caused a crash if `hid_enumerate()` returned info for a device without a serial number.
27. Fixed a bug caused by a conflict with a definition of **IsLoggingEnabled** in the Windows include file `UrlMon.h`.
28. Changed several `U2A_ApiClass` member function names for consistency with the SDK function names and the documentation:

Old Name	New Name
I2C_Register_Write	I2C_RegisterWrite
I2C_Multi_Register_Read	I2C_MultiRegisterRead
I2C_Read_Internal	I2C_ReadInternal
I2C_Write_Internal	I2C_WriteInternal
I2C_Raw_Write	I2C_RawWrite
I2C_Raw_Read	I2C_RawRead
GPIO_Write_Control	GPIO_WriteControl
GPIO_Write_State	GPIO_WriteState
GPIO_Read_State	GPIO_ReadState
LED_Write_Control	LED_WriteControl

29. Fixed a bug that caused the **U2A\_ApiClass::Power\_ReadStatus** function to fail.
30. Fixed a bug that caused the **U2A\_ApiClass::I2C\_RegisterWrite** function to fail because the first parameter was 8-bits, instead of 16-bits.
31. Increased size of internal buffer for the `hexstring()` function from 128 bytes to 512 bytes.
32. Added internal `GetApiHandleFromSerialNumber()` function to allow a `U2A_HANDLE` to be obtained from the serial number of an opened board.

33. Added **u2aSetResponseMode** API function. Only **RESP\_SYNC** and **RESP\_POLLED** modes are supported at this time. This function is currently supported only for use by **USB2ANY\_API.DLL** and is not documented.

#### **Known Issues:**

34. Disabled **MultiControllerMode** for this release. This version supports only one USB2ANY controller per instance of USB2ANY.DLL.
35. The functions **GPIO\_SetPort**, **GPIO\_WritePort**, and **GPIO\_ReadPort** are not implemented in this release. They will be implemented in a future release.
36. I2C transactions at 10kHz may have problems (e.g., missing STOP) when attempting to transfer more than 12 bytes of data. It is recommended that only very small packets (i.e., less than 10 bytes) are used when the bit rate is 10kHz. This problem will be addressed in a future release.

## **v2.0.8**

1. The name of the FEC Interval enumerated constants has been changed from **FEC\_Int** to **FEC\_Interval** to prevent conflicts with other constants.
2. Added enumerated constants for **EasyScale\_WriteSpeed**, **EasyScale\_ReadSpeed**, **EasyScale\_WriteACK**, **EasyScale\_Status\_Received**, **EasyScale\_ReceivedDataStatus**, and **EasyScale\_Threshold**.
3. Changed the name of a constant from **Command\_EasyScale\_Set** to **Cmd\_EasyScale\_Control**.
4. Changed the name of a constant from **Command\_EasyScale\_Write** to **Cmd\_EasyScale\_Write**.
5. Added constant **Cmd\_EasyScale\_Read**.
6. Fixed bug that could prevent **u2aFirmwareVersion\_Read** from returning correct value.
7. **u2aGetStatusText** will now use an internal buffer if second parameter is NULL.
8. **u2aReadResponse** will now write received bytes to debug log.
9. **u2aI2C\_RegisterWrite** was erroneously exported as **u2aI2C\_Register\_Write**.
10. **u2aI2C\_RegisterRead** was erroneously exported as **u2aI2C\_Register\_Read**.
11. Fixed bug in **u2aI2C\_RegisterRead** that prevented it from returning the correct value.
12. **u2aI2C\_MultiRegisterWrite** was erroneously exported as **u2aI2C\_Multi\_Register\_Write**.
13. **u2aI2C\_MultiRegisterRead** was erroneously exported as **u2aI2C\_Multi\_Register\_Read**.
14. Fixed bug in **u2aI2C\_MultiRegisterRead** that prevented it from returning the correct value.

**NOTE: The following changes (in blue) are internal and not directly visible to end users.**

15. Added class functions **FEC\_Control**, **FEC\_CountAndRead**, **Interrupt\_Control**, **EasyScale\_Control**, **EasyScale\_Write**, and **EasyScale\_Read**.
16. Modifications for FEC functions.
17. Added code for USB plug/unplug detection (WIP).
18. Fixed bug in **u2aI2C\_InternalRead** that could cause wrong return value in **POLLED\_MODE**.
19. Fixed bug in **u2aI2C\_RawRead** that could cause wrong return value in **POLLED\_MODE**.
20. Fixed bug in **u2aGPIO\_ReadState** that could cause wrong return value in **POLLED\_MODE**.
21. Fixed bug in **u2aADC\_ConvertAndRead** that could cause wrong return value in **POLLED\_MODE**.

22. Fixed bug in **u2aMSP430\_WordRead** that could cause wrong return value in POLLED\_MODE.
23. Fixed bug in **u2aMSP430\_ByteRead** that could cause wrong return value in POLLED\_MODE.