Release Notes

BTM44X Enhanced Data Module Firmware v240

April 2012

The firmware change history is presented below in reverse chronological order. Production releases are highlighted in blue.

Build	Description
240#0	(*) 19 Apr 2012 (44XREL240) - Only version number change to 240 to productionise
235#0	(*) 11 Apr 2012 (44XDEV01) - Only version number change as 233 was sent for testing by customer
233#0	(*) 10 Apr 2012 (44XDEV01) - Changed S Reg 80 limits to 1000..75000 which is for SPP bridging latency. The lower limit is in line with the UART best latency as specified by the new S Reg 84.
231#0	(*) 3 Apr 2012 (44XDEV01) - Added S Reg 84 to specify UART latency (default, high, medium, low) - Ensures that for low latency applications (e.g. HID devices) packets are processed as soon as they are sent by the host. - Added a more generic funtion for controlling DTR called TransportSetDTR()
229#3	(*) 21 Mar 2012 - 44XENG05 Copy integrated into 44XMAIN00 - 44XMAIN00 Merge integrated into 44XDEV01 - This means 44XDEV01 now has Glucose Meter Data Specialisation
229#0	(*) 19 Mar 2012 - Added rtc_get_build_time() to TimerApi.c so that build output files do not change based on time. They only change if there is a code change.
225#5	(-) 20 Jan 2012 -- Changes arising from new 'test' stack from CCL (*) Also assigned branchId of 3 to the '44XDEV01_Post3844' branch.
225#2	(*) 21 DEc 2011 (Branch 44XENG05) (*) Added the continua cert list to the MDS table for each of the three data specialisations (*) Added the regression tests to test that the new attr added to MDScan be read from the manager and that it has the correct data
225#1	(*) 15 Jan 2011 Enhancement: Added S Reg 82 & 83 to allow setting of RTS deassert and assert thresholds.
225	(-) 20 Dec 2011 -- Changes arising from new stack from CCL (*) phdc_transport_mgt_register() is now called very early in ProfileHdpAgentInitialise() to see if the HDP unreliability can be solved.

223	(-) 19 Dec 2011 -- Changes arising from new stack from CCL (*) Makefile: added VENA_INC_PHDC_STARTUP := 0 (*) Misc Files : Pairing has been restructured by CCL (*) Added SReg 244 (0xF4) used to allow any custom baud in range 1200 to 921600. (*) In AT mode remapped S Reg 521 so that it now points to MP Sreg 244 (*) Bug Fix: NvInfoTrustedDb.c --> sizeof() was used instead of WordSizeOf() that resulted in Win32 build not being able to read and write to the trusted device database. (*) CCL API change related changes to: - PairingManager.c - PioManager.c (*) Major significant changes to HDP related code as the CCL have made a lot of changes (*) Added ATi202 which returns the number of times DSR input has toggled. (*) Added ability to specify custom baudrate. This means there is a danger that a baudrate not recognised by a PC could be mistakenly entered and thus render the module inoperable. - ADDED code so that if on power up DSR is deasserted AND in the first 1.5 seconds, the DSR input toggles for more than 8 times, then the comms parameters will be changed temporarily until power down, to 115200,N,8,1
223	(*) 19 Nov 2011 (Branch 44XENG05) (*) Enhancement: Added Glucose Meter Data Specialisation (*) Added HdpAgentGlucometer.c to the ProfileHspAgent makefile (*) Added case MDC_DEV_SPEC_PROFILE_TEMP (4113) to ProfileHdpAgentPhdcBind() (*) Visbility: Formally made Thermomter Specialisation available (*) HDPAGENT_ENABLE_BATTERY_POWER is defined to expose battery attibutes in all data specialisations. (*) Added the MDC_ATTR_TIME_PD_MSMT_ACTIVE attribute to the Thermometer spec (*) Added Battery Charge attributes to Weigh Scale specialisation (*) Moved power status and battery charge variables to ProfileHdpAgent.c (shared by all specialisations)
221	(*) 5 Oct 2011 Bug Fix: First seen with HID connections when MITM pairing was used. Each pairing results in a key type (authenticated or non-authenticated) that needs to be stored along with the link key. The trusted device database was enhanced and each trusted device in pskey has an extra byte for this new information. AT+KY (and related MP code command) has also been updated in a backward compatible way. Symptom: Before the fix, you would pair (with MITM enabled) with S Reg 6 set to something other than 12. Then when trying to connect there would be another pairing challenge. This is because the link key type was always being hard coded as 'legacy type' without any infromation about authentication. (*) S Reg 47 now takes values 0..2, instead of 0..1. This is to allow for legacy AT+KY<addr>? use so that when this register is set to 1 then the link key type is NOT sent in the response. It has to be set to 2 for the lin key type to be appended to the response (*) AT+KY command enhaved so that it takes an optional <,value> field which is used to specify the key type when manually adding a link key to the database. (*) In MP command CMD_TRUSTED_DB_ADD, one of the rfu[4] bytes is now used to convey key type (*) In the EVT_LINK_KEY_EX event, the first byte of the rfu[4] field now contains the link key type. (*) Updated BmHostProtocol.h to refelct the new rfu[0] usage in CMD_TRUSTED_DB_ADD and EVT_LINK_KEY_EX
220	(*) Production release (28 Sep 2011)
219	(*) Skipped

217	(*) Skipped
215	(*) Added 'signed' prefix to typedefs in BmHostProtocol.h (*) Changed default value of S Reg 12 from 10 to 20 (*) Added IAS_PARAM_LINK_SUPERVISION_TIMEOUT(SECONDS_TO_625US(NvInfoSRegGetLinkSupervisionTimeoutSec())), to sp_pee_ias_register() and sp_ppe_ias_register() so that link supervision timeout works on SPP connections rather than always stack default of 20 seconds.
213	(*) Skipped
211	(*) Bug Fix: With 504=1, "NO CARRIER" was being sent to the host on disconnection.
210	(*) Production release
207	(*) Added S Reg 47 to enable reading of link key from trusted device database (*) AT mode - added AT+KY<addr>? to allow link key to be read (*) MP Mode - added EVT_LINK_KEY_EX event message to send link key information to the host when CMD_TRUSTED_DB_IS_TRUSTED is processed. The event is sent before the response for that message (*) MP Mode - CMD_TRUSTED_DB_IS_TRUSTED command processing changed so that EVT_LINK_KEY_EX is conditionally sent. (*) Added ATi100 command to return the hardware platform ID (*) Enhanced CMD_INFORMATION to return same information as ATi100 in MP Mode
205	(*) AT mode inquiry state machine : fixed a race condition
203	(*) In MP Mode, Rfcomm Framesize in no longer limited to 253
201	(*) Skipped
199	(*) Enhanced flags in CMD_INQUIRY_REQ for MP so that EIR responses can be requested. And in this case since EIR data length is NOT fixed, the inquiry results will be sent in a dedicated data channel number CHANNELID_MISC_EIR_INQ_RESP instead of the EVT_INQUIRY_RESULT event.
197	(*) Bug Fix: If connecting to a legacy pairing device that has not been paired then CMD_PINCODE does not get through to the stack because the parser is waiting for the response to the connection make command to release the parser. (*) Added new MP confirmation packet CNF_PINCODE which does the same as CMD_PINCODE, but there is no response. Also because it is not a command packet, the packet gets processed immediately. Now the host sends CNF_PINCODE to complete the pairing process and allow the connection to happen. (*) Memory Leak Bug Fix: All AT commands that use the function AtParser_PeekSpliceDeEscape_String() and there is a syntax error because of extra parameters in the line will have resulted in memory leak. (*) Added capability to specify Enhanced Inquiry Response (*) Enhanced Blob Manager with new SubCommandID in the range 7 to 9 inclusive for managing Enhanced Inquiry Responses (*) Added AT+BTB="string" (*) Added AT+BTB+"string" (*) Added AT+BTBnnnn (*) EIR data is now stored in PSKEY User41 so that it can be used on power up to change the default EIR string. (*) Added AT+BTIE variant of the inquiry request command which will result in enhanced inquiry response where RSSI and EIR data is appended to the expected response. However this time, devclass is automatically enforced and in place of the friendlyname it just sends a " ", place holder.

195	(*) Sniff mode parameters are now displayed via ATI13. Only displays the four sniff parameters. 1st line in milliseconds and second in appropriate slots. Note slot calculation is different for first two (in units of 0.625ms and second two are in 1.250ms) (*) Added function NvInfoManagerGetSniffParameters() (*) Bug fix (maths overflow) when converting sniff msec to slots in function LibManagerTimeMsTo625usSlots() (*) Changed range of SReg73 (alias 561) to 0..40000 from 0..20000
193	(*) Bug fix : SReg 7,8,9,10 values 10 to 19 did not work because out of spec due to rounding down to 10 and spec says minimum value should be 10.625 ms (*) Changed range of SReg 7,8,9,10 to 12..2560 from 10..2550
191	(*) No changes
190	(*) Production build
189	(*) Ability to store pincode using AT+BTK in non-vol memory when a pairing is not ongoing.
187	(*) Added S Reg 507 to enable/disable DSR check when in AT mode (*) Added S Reg 651 to 654 (rfcomm modem sig mapping to gpio) (*) Added S Reg 661 to 664 (rfcomm modem sig mapping to gpio)
185	(*) Enhancement: S Reg 255 now accepts 0 to allow protocol to be selected via a gpio pin. (*) SReg 50 to 65 take max value of 15 now to cater for the protocol mode in and out.
184	(*) Production release

global solutions: local support™
USA: +1.800.492.2320
Europe: +44.1628.858.940
Asia: +852.2268.6567
wirelessinfo@lairdtech.com
www.lairdtech.com/wireless

Laird Technologies is the world leader in the design and manufacture of customized, performance-critical products for wireless and other advanced electronics applications. Laird Technologies partners with its customers to find solutions for applications in various industries such as:

- Network Equipment
- Telecommunications
- Data Communications
- Automotive Electronics
- Computers
- Aerospace
- Military
- Medical Equipment
- Consumer Electronics

Laird Technologies offers its customers unique product solutions, dedication to research and development, as well as a seamless network of manufacturing and customer support facilities across the globe.