Release Notes

BTM41X Firmware version 16.1.3.0

August 2011

Prepared by :	Sebastian Schillack	**Authorised by:**	Andrew Dobbing
Signature :		**Signature :**	

Change History

Issue	Change	Author	Date
001	v16.1.0.5	Sebastian Schillack	31 Jul 09
002	V16.1.1.0	Sebastian Schillack	19 Aug 09
003	V16.1.2.0	Sebastian Schillack	18 Nov 09
004	V16.1.2.1	Sebastian Schillack	06 July 10
005	V16.1.2.2	Sebastian Schillack	24 Feb 11
006	V16.1.2.3	Sebastian Schillack	02 Mar 11
007	V16.1.2.4	Sebastian Schillack	07 Apr 11
008	V16.1.2.5	Sebastian Schillack	11 Apr 11
009	V16.1.2.6	Sebastian Schillack	28 Apr 11
010	V16.1.2.8	Sebastian Schillack	15 July 11
011	V16.1.2.9	Sebastian Schillack	16 Aug 11
012	V16.1.3.0	Sebastian Schillack	28 Oct 11

1 FIRMWARE VERSION 16.1.0.5

1.1 New features

- Bluetooth 2.1 Supported
- Serial port profile (SPP)
- SCO/eSCO audio connection (external codec board required)
- Sniff mode
- Standard / Non-standard UART baud rates configurable
- Secure simple pairing (SSP)
- CSR Bluestack build 5361 (native mode, 56bit encryption, 23e)

1.2 Changes in AT command set

- AT+BTIE replaced by AT+BTE
- New inquiry commands:
  - AT+BTIR: inquiry responses with RSSI data
  - AT+BTIE: inquiry responses with extended inquiry response data
- New commands for Serial Port Profile (SPP):
  - AT+SPD<BdAddr>: initiate SPP connection
  - AT+SPH : release SPP connection
  - The commands "ATD<BdAddr>" and "ATH" still exist but are not recommended for future usage. Background: The old command's syntax was not in line with the syntax of all remaining profile related commands. All profile related AT commands are now moved to a higher layer of abstraction based on top of the CSR profile libraries (represented by each profile's own

namespace in the AT command set). The old commands were on a lower layer of abstraction. A new low level AT command for RFcomm connections is planned for the future.

- <UUID> option for ATD<BdAddr> deprecated

1.3 Bug Fix

- Significantly lower inquiry performance (compared to BISM2) was caused by missing initialisation of S registers 541 and 542, fixed

1.4 Known Issues

- AT+BTE? doesn't respond correctly
- GPIOs not accessible by AT-commands
- V16.1.0.5 has not been tested by automated regression tests, hence there might be unknown bugs left with this version

2 Firmware Version 16.1.1.0

2.1 New features (compared to 16.1.0.5)

- ATI19 : SCO connection state
- AT+BTA responding with ERROR14 if in wrong state

2.2 Changes in AT command set (compared to 16.1.0.5)

- None.

2.3 Bug Fix

- Automated regression tests: minor bugs fixed, related to SSP, EIR,
- "AT+BTE?" fixed
- "AT+GOU"/"AT+GIU": end of range error appeared one step below actual end of range, fixed

2.4 Known Issues

- GPIOs not accessible by AT-commands, Ref. [1-10]
- General Bonding (automated pairing during connection setup) with legacy devices
- (BT2.0 or earlier) does not work. Ref. [1-2]
  Workaround: use AT+BTW<BdAddr> to initiate dedicated bonding prior to connection establishment with legacy devices.
- General Bonding (automated pairing during connection setup) with MITM (Man In the middle protection) enabled does not work. Reason: no input data is accepted by the initiating device until the connection is established. Ref. [1-3]
  Workaround: use AT+BTW<BdAddr> to initiate dedicated bonding prior to initiation of a connection with MITM enabled.
- S324 is ignored, SSP timeout is fixed to 30s (probably error in CSR library) Ref. [1-7]

3 Firmware Version 16.1.2.0

3.1 New features (compared to 16.1.1.0)

- None.

3.2 Changes in AT command set (compared to 16.1.1.0)

- AT+BTD* deletes the entire trusted device list plus the cashed link key.
- AT+BTDW deletes the cashed link key only.

3.3 Bugs Fixed

- DSR-toggle issue: when S507=1 or 2, a de-asserting/re-asserting the DTR/DSR- line within the time specified by S519 caused a disconnection rather than changing from "connected" to "command and connected" mode. This issue has been fixed. Ref.[1-11]
- Loss of link: module was not controllable any more after a loss of link. This issue is fixed. Ref. [1-13]
- After a SPP link was released, the module was not discoverable any longer although discoverability was enabled. Due to the character of this bug, issues related to discoverability and connectability were likely to occur in other scenarios as well. This bug is now fixed. Ref. [1-15]
- After a command sequence of AT+BTQ,AT+BTX,AT+BTG the module should have been connectable only but in fact it was discoverable too. This issue is fixed. Ref. [1-5]
- After a command sequence of AT+BTG,AT+BTX,AT+BTQ the module was turning discoverable and connectable, rather than the intended discoverable. This has been fixed. Ref. [1-6]
- S registers 589 and 590 were not initialized to their default values on AT&F*, this has been fixed. Ref. [1-14]
- New range for S531 is [0..4]. Value of 5 removed because Daemon-mode is not available on this firmware. Ref. [1-17]
- AT+BTW? still displayed a cashed link key although it was deleted by AT+BTD*. This has been fixed. AT+BTD* now generally deletes the trusted device list AND the cached link key. AT+BTDW deletes the cashed link key only. In earlier versions, the cached link key was only deleted on power cycle. Ref. [1-12]
- General Bonding (automated pairing during connection setup) with MITM (Man In the middle protection) enabled was not working. Reason: No AT commands were accepted by the initiating device until the connection was established or until the NO CARRIER message was displayed. This has been fixed. The parser becomes temporarily enabled whenever user input is required, e.g. on "PAIR ? …", "PASSKEY ?" or "PIN ?". In this state the only AT commands accepted are: AT+BTBY, AT+BTBN, AT+BTB<6-digit- passkey> and AT+BTK="<PIN>" Ref. [1-3]
- Range query error: "ATS<x>=?" has returned ERROR 01 for S520…S525 and S1001..S1010, this has been fixed Ref. 1-9]

3.4 Known Issues

- GPIOs not accessible by AT-commands, Ref. [1-10]
- In some cases, General Bonding (automated pairing during connection setup) with legacy devices (BT2.0 or earlier) does not work. It did not work when tested against a BISM2. However it was working well when tested against a BTM410 that was converted to BT2.0. This implies that this issue is likely caused by the remote device. Ref. [1-2]
  Workaround: use AT+BTW<BdAddr> to initiate dedicated bonding prior to connection establishment with a legacy devices.
- S324 is ignored, SSP timeout is fixed to 30s (probably error in CSR library) Ref.[1-7]
- Connection to Nokia E71 does not work, Ref.[1-4]
- When the link key for a device exists locally (AT+BTT? or AT+BTW?) but the key is missing in the remote device (e.g. it has been deleted there), creating a connection (AT+SPD<BdAddr>) will result in NO CARRIER rather than new pairing. This is caused by the underlying bluestack and must be resolved in a future update. The first number of the firmware version (16 here) indicates the bluestack version in use. So in every firmware version beginning with '16' this issue will be present. Ref. [1-21]
  Workaround (a) (recommended): use AT+BTW<BdAddr> to initiate new pairing to generate new link

keys on both devices.
Workaround (b): the link key for the remote device can be deleted from the trusted device list (AT+BTD<BdAddr>) or from cache (AT+BTDW). A new connection attempt (AT+SPD<BdAddr>) should initiate new pairing automatically. However, this is likely not to work with some legacy (BT2.0 and earlier) devices. Hence workaround (a) is recommended.

4 FIRMWARE VERSION 16.1.2.1 (ENGINEERING FIRMWARE)

4.1 New Features (compared to v16.1.2.0)

- support for extended inquiry response (EIR) data enhanced, enabling up to 240 Bytes of EIR data
- added S544

4.2 New AT commands

- AT+BTE : clear EIR data from baseband
- AT+BTE? : query EIR data from baseband
- AT+BTE="<data>" : write EIR data to RAM buffer and baseband
- AT+BTE="" : delete EIR data from RAM buffer and baseband
- AT+BTE+"<data>" : append EIR data to RAM buffer
- AT+BTE~ : copy EIR data from RAM buffer to baseband
- AT+BTE+? : query EIR RAM buffer
- AT+BTE+"" : clear EIR RAM buffer
- AT+BTEW : copy EIR RAM buffer to persistent store
- AT+BTED : delete EIR data from persistent store
- AT+BTEW? : query EIR data from persistent store
  Note: the content of the persistent EIR data storage is copied to the baseband at boot time

4.3 New S-Registers

- S544 : configure UART for high throughput or low latency:
  - 0 = low latency
  - 1 = high throughput (default)
  - A new setting needs subsequent "AT&W" (store value) and "ATZ" (reset) to take effect.

4.4 New Error Codes

- 76: memory allocation attempt was unsuccessful

4.5 General Information

This release is engineering firmware and must be used for development only. The new features of this release are not covered yet by automated regression tests. There have been plenty of manual sanity checks but there might still be bugs and untested use cases left.

4.6 Known Issues

- See section 3.4.

5 Firmware Version 16.1.2.2 (engineering firmware)

5.1 General information

This firmware is an engineering release for a particular project. It contains two bug fixes (BTX/ATO and SPP malloc issue) and two new ATI commands (ATI27, ATI1000).

This firmware contains additional new features in terms of GPIO and EIR data, but they should be ignored since the implementation of these features is unfinished. They will be covered by the next general release.

This release is engineering firmware and must be used for development only. The new features of this release are not covered yet by automated regression tests. There have been manual sanity checks but there might still be bugs and untested use cases left.

5.2 New ATI commands

- ATI27: query current scan state
  - 0 = not discoverable and not connectable (not scanning)
  - 1 = discoverable (inquiry scanning)
  - 2 = connectable (page scanning)
  - 3 = discoverable and connectable (inquiry- and page-scanning)
- ATI1000: number of available memory slots
  - This diagnostic command helps to trace internal memory allocation issues of the module. If the number returned by ATI1000 nears zero, an automatic reset of the module is likely to occur due to lack of free memory slots.

5.3 Bugs fixed (compared to 16.1.2.1)

- When issuing AT+BTX in command and connected mode (SPP) and then issuing ATO, ERROR 04 was returned. This prevented the return to data and connected mode (SPP). This issue is fixed. [Ref. 2-14]
- On the 9th SPP connect attempt after 8 SPP connect/disconnect cycles, the module went through a reset automatically. This could have created issues because this behaviour is probably not expected by the host controller. However, this issue is fixed. [Ref 2-13]

6 Firmware Version 16.1.2.3 (engineering firmware)

6.1 General information

This firmware is an engineering release for a particular project. It contains one bug fix (“PASSKEY N …” 6 digit/leading zeroes issue).

Beyond that, this firmware contains additional new features in terms of GPIO and EIR data, but they should be ignored with this release since the implementation of these features is unfinished. These features will be completely covered by the next general release.

This release is engineering firmware and must be used for development only. The new features of this release are not covered yet by automated regression tests. There have been manual sanity checks but bugs and untested use cases may remain.

6.2 Bugs fixed (compared to 16.1.2.2)

- Leading zeroes were not displayed in the passkey of the following asynchronous messages:
  PAIR ? <BdAddr>, “<friendlyname>”,<Passkey> PASSKEY N <BdAddr>,”<friendlyname>”,<Passkey>
  Therefore it was possible that the passkey contained less than 6 digits. This is now fixed. Leading zeroes are now inserted so a 6 digit passkey is guaranteed in any case. [Ref 2-15 / 2-12]

7 Firmware Version 16.1.2.4 (engineering firmware)

7.1 New Features (compared to v16.1.2.3)

- GPIOs (general purpose input/output) support:
  - 4 freely usable GPIOs available
  - Function mapping (e.g. GPIO cable replacement, volume up/down)
- Improved support for extended inquiry data (EIR)

7.2 New S-Registers

- S650: GPIO pin state mask
  - 0 = configuration field enabled, 1 = pin state mask for S651..S658
- S651..S658: GPIO configuration registers, for details see below "GPIO Configuration"
- S669: GPIO input strong bias enable bitmask
  - 1 = enable strong internal pull up/down if GPIO is an input
  - 0 = enable weak internal pull up/down if GPIO is an input
- S670: read/write pin states of all GPIOs in one step
- S411: short press duration in (ms), granularity 200 ms
- S412: medium press duration in (ms), added to value of S411, granularity 500 ms
- S413: long press duration (ms), added to values of S411+S412, granularity 500 ms

7.3 New ATI commands

- ATI27: query current scan state
  - 0 = not discoverable and not connectable (not scanning)
  - 1 = discoverable (inquiry scanning)
  - 2 = connectable (page scanning)
  - 3 = discoverable and connectable (inquiry- and page-scanning)
- ATI29: query maximum data length for EIR data
- ATI30: query current RAM buffer length (EIR)
- ATI31: query current baseband buffer length (EIR)
- ATI411: short press duration (time in ms)
- ATI412: medium press duration (absolute time in ms, S411+S412)
- ATI413: long press duration (absolute time in ms, S411+S412+S413)

7.4 New Error Codes

- 79: writing to modem control line is not permitted by GPIO S-register
- 80: attempting to write the pin state of a GPIO that is configured as input
- 81: Maximum size of EIR data exceeded (ATI29)

7.5 GPIO configuration

A GPIO pin is configured by S651…S658 according to the following bit mapping:

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	function mapping code								reserved		FMS		NEN	INV	DIR	PS
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Field	Description
0 – PS	I/O Pin State – returns the current state of the I/O pin (0/1) A write affects the I/0 pin directly if DIR=1 and FME=0
1 – DIR	Direction – controls if the I/O pin is an input or an output 0 input, 1 output
2 – INV	Inversion – controls if the I/O pin is inverted or not. Applies to both pin directions (read and write). 0 not inverted , 1 inverted
3 – NEN	Notification enable – sends a notification to the host via UART on each status change. Applies to both pin directions (input and output). AT&W and ATZ is required for a new setting to take effect. Notification format: "\r\nGPIOx,y\r\n" x=GPIO index [1..8]; y=new pin state [0..1] with INV flag applied 0 disable notification, 1 enable notification
4 – FMS	Function Mapping Select – Selects modem control function or Function Mapping Code 0 FMS_NONE – function mapping disabled 1 FMS_MODEMCTRL – use GPIO as modem control line (1) 2 reserved (attempt to write this value will cause ERROR 02) 3 FMS_MAPCODE – bits [8:15] specify Function Mapping Code
[5:7]	Reserved
[8:15] function mapping code	A Function Mapping Code is set in this bit field. The pin will carry out the assigned operation. A mapped function does not affect any other flag of the GPIO configuration register. So, e.g. DIR and INV must be set manually according to the selected function and hardware requirements. Valid only if FMS==3. FMS selects function mapping.
(1): read only	

Value	Function Mapping Code – Input
0x00	Cable Replacement TX
0x01	RFC_RTC_TX(1)
0x02	RFC_RTR_TX(1)
0x03	RFC_IC_TX(1)
0x04	RFC_DV_TX(1)
0x05	Volume down single step(2)
0x06	Volume up single step(2)
0x07	Volume down multiple after short press(2)
0x08	Volume up multiple after short press(2)
0x09	Volume down multiple after medium press(2)
0x0A	Volume up multiple after medium press(2)
(1): disabled, reserved for future usage (2): applies to SCO connection with external codec board	

Value	Function Mapping Code – Output
0x00	Cable Replacement RX
0x01	RFC_RTC_RX(1)
0x02	RFC_RTR_RX(1)
0x03	RFC_IC_RX(1)
0x04	RFC_DV_RX(1)
(1): disabled, reserved for future usage	

7.6 EIR data

The block diagram below gives an overview of the "AT+BTE" command family which has become more stable with this firmware version.

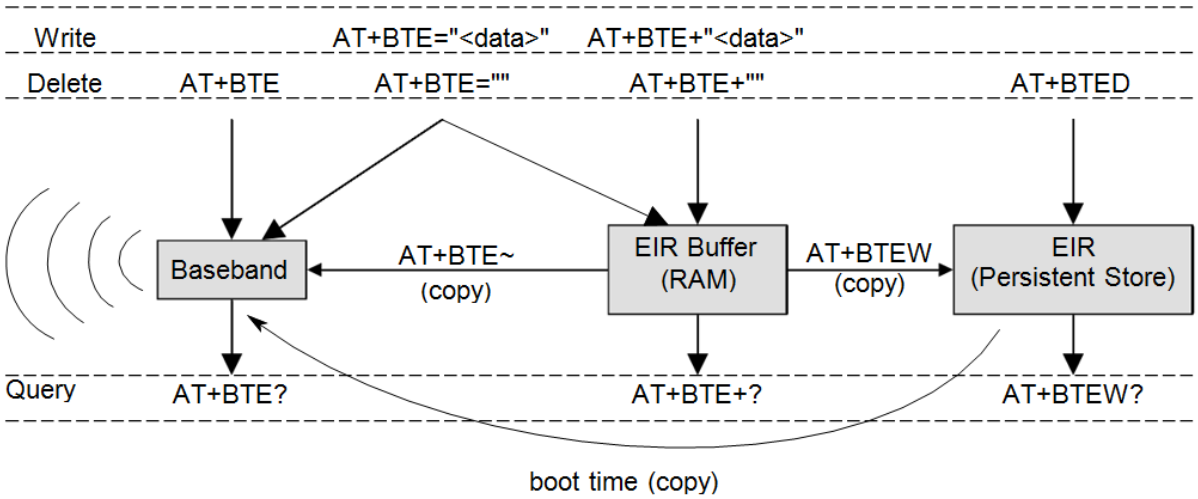

Figure 1: BT2.1 - Extended Inquiry Response - "AT+BTE" command family

7.7 Bugs fixed (compared to 16.1.2.3)

- With S504=1, "CONNECT" was not suppressed on ATO, this has been fixed [Ref 2-9]
- General: Some S-Registers (300..306, 310..313) were available but were not related to profiles of the BTM41x. These S-Registers have been removed [Ref 2-11]
- Discoverable/Connectable: S555=3 did not work: after the initial time window (configured by S554), the module always went back to non-discoverable and non-connectable state, regardless of S555. This has been fixed, S555 should work as expected now. The discoverable/connectable status can be checked by ATI27. [Ref 2-16]
- General: When a peer device enables sniff sub-rating, the Bluetooth device address of that device was printed to the UART unexpectedly. This has been fixed. [Ref 2-18]
- "959 Byte limit": when transmitting data to BTM41x (SPP) over the air from a PC via the built-in Bluetooth device or a BT-USB dongle, the data block for one write command was limited to 959 bytes. Any data beyond 959 bytes that was not forwarded to the UART of BTM41x. The missing data was conveyed on the next write command. This issue has been fixed. [Ref. 2-19]
- Inquiry: When inquiring with friendly name (e.g. AT+BTIN) and a remote device was discoverable but not connectable, this remote device was not listed in the inquiry results though discoverable. This has been fixed now. If the friendly name cannot be retrieved (e.g. due to not connectable), then a '!' will be displayed instead of the friendly name. [Ref 2-20]

7.8 General information

The new features described for this firmware version (v16.1.2.4) have not yet been tested by automated regression tests. This will be done before the release of the next production firmware for BTM41x.

7.9 Known issues

- 4096 byte limit on development kit: When sending data to the COM port (FTDI USB2Serial converter) of a BTM41x development kit, the data size of one WriteFile() command must not exceed the USB2Serial converter buffer size which is setup in the driver properties. A typical value is 4096 bytes (which is default and maximum). Any data beyond that limit is lost if written in one write operation. As a workaround, data must be split to chunks of less or equal the buffer size and written with several WriteFile() calls. This is not caused by the BTM41x itself but by the USB-to-serial converter of the development kit.

8 Firmware Version 16.1.2.5 (engineering firmware)

8.1 General information

This firmware is an engineering release for a particular project. It contains a bug fix for a SPP data transmission issue ("smart disconnect"). Please find the issue description below.

This release is engineering firmware and must be used for development only. The new features of this release are not yet covered by automated regression tests. There have been manual sanity checks but bugs and untested use cases may remain.

8.2 New S-Registers

- S319: smart disconnect
  - 0 = feature disabled
  - 1 (default) = smart disconnect enabled
    With this setting, BTM41x tries to detect if there is any data pending in its internal buffers on an incoming disconnect notification. If so, then the BTM41x delays the disconnection until all pending data has been delivered to the UART first and then signals the disconnection on the UART ("NO CARRIER") and on the DCD line.
  - This is an experimental feature which may have side effects in certain situations. If issues are observed, then this feature should be disabled.

8.3 Description of the issue

The issue that this release fixes occurs in the following scenario:

Hardware setup:

- PC -> BT-USB dongle / built in BT device -> SPP link over the air -> BTM41x -> UART

Software setup:

- BTM41x has been discovered by the PC, has been paired and a COM-port number has been assigned on the PC for the BTM41x
- As soon as the assigned COM port is opened on the PC (e.g. by a terminal program or another application), the SPP link to the BTM41x is established automatically, initiated by the PC.
- A C/C++ application utilizes the COM port in the following manner:
  handle = CreateFile (…); //opening COM port and establishing SPP link
  WriteFile(…); //write (send) data
  CloseHandle(…); //closing COM port, disconnect SPP link

Issue:

- It has been observed that some remainder of the data seems to be lost and not being transmitted to the BTM41x UART. On the next open/send/close attempt, the SPP link becomes unstable and hangs. A reset of the BTM41x helps and after a timeout the WriteFile(…) function returns.

  The issue depends on two parameters:

  1. The size of the data being transmitted with one write command, and
  2. The baud rate of the BTM41x UART.

  Furthermore, the issue can be prevented by using a delay (e.g. "Sleep(…)") before "CloseHandle(…)"

Solution / workaround:

On PC:

- Insert a delay before disconnecting and closing the COM port (CloseHandle(…))

On BTM41x:

- Set S319=1 (enable smart disconnect). This improves the condition but does not solve it. Several consecutive manual tests have shown the following parameters seem to be stable:
  - Delay before closing COM port: no
  - BTM41x BaudRate (S520): 115200
  - Data size: 4480

Increasing data size or decreasing the BaudRate have resulted in the issue to re-occur in the test environment.

Different PC Bluetooth stacks (e.g. Toshiba BT stack vs. Microsoft BT stack) may also yield different results.Therefore, you must experiement to determine the optimal parameters for the a final application if utilizing the WriteFile(…) / CloseHandle(…) approach.

9 Firmware Version 16.1.2.6 (engineering firmware)

9.1 General information

This firmware is an engineering release. It contains a bug fix for a DCD signalling issue which occurred when connecting to a PC via serial port profile (SPP), as well as some improvements related to legacy pairing (PIN code based, pre-BT2.1)

Engineering firmware must be used for development only. The new features / bug fixes of this release are not covered yet by automated regression tests. There have been manual sanity checks but there may still be bugs and untested use cases.

9.2 New S-Registers

- S356: simple default PIN code (legacy pairing)
  - 0 (default) = feature disabled
  - 1 = "0000"
  - 2 = "1234"
  - 3 = "8888"

  This register is only referenced on a PIN code request when no persistent PIN code is stored in the module (see ATI59). It caters to simple commonly used PIN codes by many BT2.0 devices. If the feature is disabled (value=0) and no persistent PIN code is available, then the PIN code request will be forwarded to the module's host by the asynchronous "PIN?" message.

9.3 New ATI commands

- ATI59: query if a persistent PIN code (legacy pairing) has been set (by AT+BTK="…")
  - 0 = persistent PIN code not stored
  - 1 = persistent PIN code stored

  A persistent PIN code can be deleted by AT+BTK="". The PIN code itself is not presented for security reasons.

9.4 Bugs fixed (compared to 16.1.2.5)

- DCD issue on PC side: When initiating an SPP (serial port profile) connection from a PC to the module, the DCD line was not asserted when entering the link on PC side (terminal program, e.g. EzurioTerminal). This issue is fixed. However, it still persists when initiating from the module side. [Ref. 2-22]

  This bug occurred for the first time in v16.1.2.1.

9.5 Other changes

- S319: factory default value = 0 (smart disconnect disabled, see version 16.1.2.5)

10 Firmware Version 16.1.2.8 (Engineering F/W)

10.1 New S-Registers

- S358: simple default PIN code (legacy pairing)

This S-Register is identical to S356 (see Section 9.2 above). S356 is already mapped to a different function on the BTM5xx series. S358 was created to maintain consistency between the S-registers of BTM41x and BTM5xx series. Laird recommends you use S358 instead of S356.

10.2 Bugs fixed (compared to 16.1.2.6)

- DCD issue on PC side: When making an SPP connection from the module to a PC, the DCD line was not asserted on the PC side when entering the link (terminal, e.g. EzurioTerminal). This issue is fixed [Ref. 2-23].

10.3 Other

- Firmware version 16.1.2.7 was built but not released. This explains the skip in the version numbering.

11 Firmware Version 16.1.2.9 (Engineering F/W)

11.1 New S-Registers

- S359 : Auto-BTX on SPP connection [0..3], default = 3:

  - 0 = do not change discoverable and connectable state when entering or exiting an SPP connection
  - 1 = Set the module to not discoverable / not connectable when entering an SPP connection (Bit 0)
  - 2 = restore discoverable/connectable state to S512 settings when exiting an SPP connection (Bit 1)
  - 3 = combination of 1 and 2

Notes: 1. Inquiry scanning and page scanning (discoverable and connectable state) consume power and bandwidth. Bit 0 of this S-Register has the same effect as issuing the following sequence after an SPP connection has been established (assuming data mode):

```
^^^             # changing from data to command mode
AT+BTX\n        # make not discoverable, not connectable
ATO\n           # revert to data mode
```

This is the reason why this function is named "Auto-BTX".

2. The current scan state can be queried by ATI27.

12 FIRMWARE VERSION 16.1.3.0 (PRODUCTION F/W)

This firmware does not provide new features. It has the same features as 16.1.2.9.