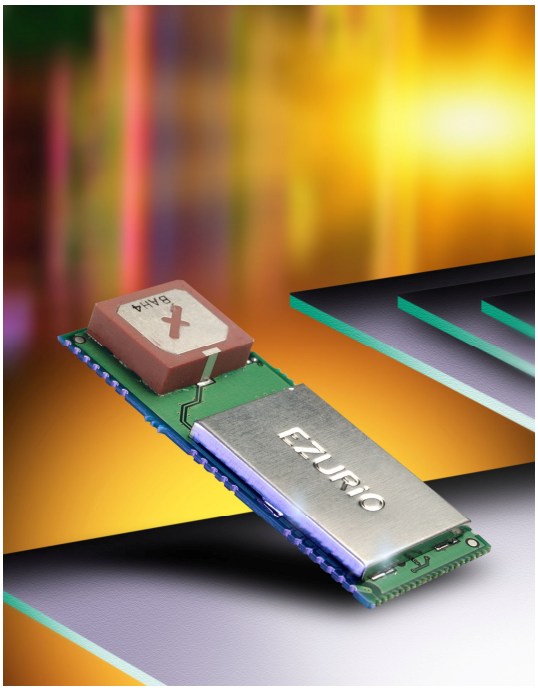
EZURiO

EZURiO



Bluetooth Module Application Scenarios

EZURiO

Example 1: Light Switch Controller

Background

Consider a small microcontroller controlling a single light switch. The microcontroller's serial port is connected to the Module and a digital output line drives a relay which controls a light. The remote host sends the text "ON" to switch on the relay and "OFF" to switch it off.

AT Command Sequence

AT Command	Response	Comment
ATZ	OK	Resets the device and sets all S Register values as per the values in non-volatile memory database.
ATS0=1	OK	Auto answer incoming connection after 1 ring
AT+BTP	OK	Will accept incoming connection from any device and will also be discoverable
	CONNECT 123456789012	Call has been answered automatically, and the Bluetooth address of the peer device is provided.
At this point the light switch protocol takes in commands from the remote. For example ON, OFF etc		
	NO CARRIER	If the remote end drops the connection
OR		
^^^	OK	Puts the device in to command mode
ATH	OK	Drops the call.

EZURiO

Example 2: Remote Data Logger

Background

A data logger on a remote site gathers data which is then transferred to a central site via a Bluetooth enabled phone. Assume that the remote site is accessed via the telephone 02089388609 and the PC connected to the modem logs the data to a file, so all the data logger has to do is open a connection and then send textual data corresponding to the gathered data.

Assume that the data logger knows the Bluetooth address of the phone and assume it is 123456789012

Further, assume that the Bluetooth Phone exposes a serial port profile through which an AT modem can be accessed.

AT Command Sequence

AT Command	Response	Comment
ATZ	OK	Resets the device and sets all S Register values as per the values in non-volatile memory database.
ATD123456789012	CONNECT 123456789012	
ATZ	OK	The response is coming from the Bluetooth phone
ATD02089388609	CONNECT	
The data logger sends data which is captured at the remote end.		
+++	OK	Puts the Bluetooth Phone Modem into command mode
ATH	OK	Drops the call to 02089388609
^^^	OK	Puts the device into command mode
ATH	OK	Drops the Bluetooth connection

EZURiO

Example 3: Trusted Device Management

Background

Bluetooth provides for secure connections through the use of link keys. The link keys are 128 bit entities which are uniquely created for each Bluetooth device via the device's Bluetooth address. Since all Bluetooth addresses are unique, this results in unique keys. The link keys are 'private' objects and as such are never exposed via the AT interface. At all times they remain within the Module. Therefore the module allows link key management by using Bluetooth addresses as handles. The device has a database of link keys. Each record in the database has two fields. One field contains the Bluetooth address and the other contains the 128 bit link key. **Only the address field is viewable**.

The following subsections describe AT command sequence for typical pairing functions.

Obtain a new Link Key

This process will obtain a new link key for a remote device whose address is 123456789012 and whose pin code is known to be 12345.

AT Command	Response	Comment
AT+BTW123456789012	OK	
	PIN? 123456789012	This is sent to the host every 2000ms
	PIN? 123456789012	
AT+BTK="12345"	OK	Pairing complete. The link key is stored in a volatile cache
	PIN 0 123456789012	Pairing successful

Obtain a new Link Key and autosave

This process will obtain a new link key for a remote device whose address is 123456789012 and whose pin code is known to be 12345 and will auto save the link key to the database.

AT Command	Response	Comment
ATS538=1	OK	
AT+BTW123456789012	OK	
	PIN? 123456789012	This is sent to the host every 2000ms
	PIN? 123456789012	

AT+BTK=”12345”	OK	Pairing complete. The link key is stored in a volatile cache
	PIN 0 123456789012 00	Pairing successful and saved automatically

Obtain a new Link Key and autosave fails

This process will obtain a new link key for a remote device whose address is 123456789012 and whose pin code is known to be 12345 and will auto save the link key to the database.

AT Command	Response	Comment
ATS538=1	OK	
AT+BTW123456789012	OK	
	PIN? 123456789012	This is sent to the host every 2000ms
	PIN? 123456789012	
AT+BTK=”12345”	OK	Pairing complete. The link key is stored in a volatile cache
	PIN 0 123456789012 22	Pairing successful and database is full

Store link key in trusted device database

This process assumes that the procedure described in “Obtain a new link key” has been done, a link key exists in the cache and that the database is not full.

AT Command	Response	Comment
AT+BTT	OK	

List trusted device database

This process assumes that the procedure described in “Obtain a new link key” has been done, a link key exists in the cache and that the database is not full.

AT Command	Response	Comment
AT+BTT?	123456789012	
	123456789013	
	123456789014	
	123456789015	
	123456789016	
	OK	

Store link key in trusted device database (full)

This process assumes that the procedure described in “Obtain a new link key” has been done, a link key exists in the cache and that the database is full.

AT Command	Response	Comment
AT+BTT	ERROR	The data base is full
ATI6	8	The maximum size of trusted devices database
	OK	
AT+BTT?	123456789012	
	123456789013	
	123456789014	
	123456789015	
	123456789016	
	OK	
AT+BTD123456789016	OK	Deletes the key associated with 123456789016
AT+BTT	OK	The store will work

Example 4: Bluetooth Enabling a Modem for DUN

Background

You have a spare external serial modem which you wish to Bluetooth enable and provide Dial-Up Networking services (DUN Profile) for PDAs or PCs within range. You also want to ensure that not anyone can use it, so that new users will need to pair to be able to use its services. The modem will be plugged into the telephone socket and the Module will be connected to the serial port of the modem.

This will allow internet access from anywhere in the home, where the client will access the DUN profile as described in the Bluetooth specification.

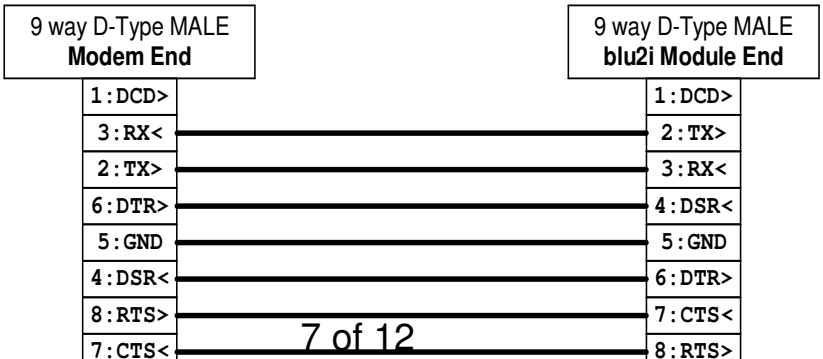
The Module will need to be prepared as per the table of AT Commands in the next section, and the Module shall be connected to the modem (assuming it has a 9 way male D type connector) using an adapter cable as per the diagram in the subsequent section.

In this configuration, the Module is acting as a proxy host to the modem.

AT Commands to set up the Module

AT Command	Response	Comment
AT&F*	OK	Clear non-volatile storage
ATS0=-1	OK	Autoanswer on 1 Ring
ATS504=1	OK	Enable Silent Operation
ATS538=1	OK	Auto save link keys in trusted device database
ATS512=7	OK	On power up, become discoverable and connectable
ATS502=1	OK	Force authentication
ATS102=5	OK	Enable SPP and DUN profiles
AT&W	OK	Save register settings to non-volatile storage
AT+BTK=”1234”	OK	Pin code to use during pairing. This can be set to any 4 digit number

Wiring between Modem and Module



EZURiO

Example 5: Modem as a remote host for a module

Background

You have a room full of Bluetooth peripherals (e.g. vending/gaming machines) which you want to remotely access via a telephone network using a modem.

This will allow a remote PC to dial into the modem which auto-answers and then subsequently it will be like having the Bluetooth module connected locally.

The serial module will need to be prepared as per the table of AT Commands in the next section, the modem will need to be prepared as per the table of AT Commands in the subsequent section and the serial module shall be connected to the modem (assuming it has a 9 way male d-type connector) using an adapter cable as per the diagram in the subsequent section.

In this configuration, the modem is acting as a proxy host to the Bluetooth serial module

AT Commands to set up the Module

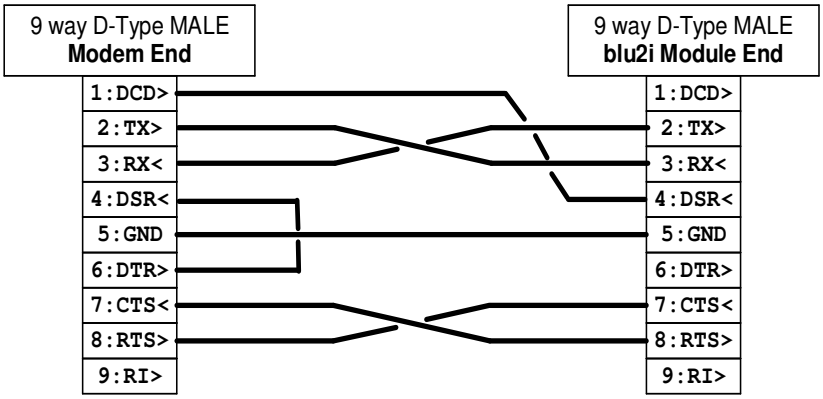
AT Command	Response	Comment
AT&F*	OK	Clear non-volatile storage
ATS512=1	OK	On power up, do not become discoverable and connectable
ATS506=0	OK	Disable echoes
ATS539=1	OK	Ignore UART RX when DSR is deasserted
AT&W	OK	Save register settings to non-volatile storage

AT Commands to set up the Modem

AT Command	Response	Comment
ATS0=1	OK	Autoanswer on 1 Ring
ATE0	OK	Disable echoes
AT&W	OK	Save register settings to non-volatile storage

EZURiO

Wiring between Modem and Module



Example 6: Auto Cable Replacement Via Pairing

Background

You have a legacy host which needs to connect to a Bluetooth device (say a mobile phone) automatically on an ad-hoc basis as a master. The master module will be configured to auto connect to a Bluetooth address using the AT+BTR command and it will assume that the device will change often. The Bluetooth address specified via AT+BTR needs to be set on an ad-hoc basis as new phones come along to offer their data bearing services.

In this case, the module is configured to accept incoming pairing attempts, and if the pairing is successful, then the address of the new device is used next for automatic connection attempts.

AT Commands to set up the Module (one-off)

AT Command	Response	Comment
AT&F*	OK	Clear non-volatile storage
ATS500=1	OK	Default authentication for outgoing connections
ATS512=1	OK	Idle mode
ATS538=1	OK	Autosave Link keys
ATS507=1	OK	Allows DSR input to be used to inhibit autoconnect cycle
ATS505=10	OK	Connection attempts timeout after 10 seconds
ATS530=15000	OK	Wait period in milliseconds between connection attempts
ATS543=1	OK	Accept incoming pairing during wait phase between connection attempts
AT&W	OK	Store new S Register settings
AT+BTK=”8938”	OK	Set Pin Code to 8938 – can be any number
AT+BTR008098000001	OK	Specify a random Bluetooth address for connection attempts. The address 008098000001 is safe to use as EZURIO have not issued a Bluetooth device with that address

Notes: If the device specified in the AT+BTR is in the neighbourhood, then it MUST be switched off to allow a pairing with a new device.

Example 7: Use the module as a headset

Background

Bluetooth headsets allow an audio connection to be established with a mobile phone. This example shows how a module mounted on a Module Development Kit motherboard can be used as a headset.

It is assumed that a terminal emulator is connected, to allow commands defined in the headset profile specification to be sent.

AT Commands to set up the Module (one-off)

AT Command	Response	Comment
AT&F*	OK	Clear non-volatile storage
ATS515=$200404	OK	Set device class as headset
ATS512=4	OK	Wait for connection
ATS538=1	OK	Autosave Link keys
ATS0=1	OK	Autoanswer on first RING
ATS102=2	OK	Enabled Headset Server profile only
ATS532=1	OK	Switch on audio channel on connection
AT&W	OK	Store new S Register settings
AT+BTK=”8938”	OK	Set Pin Code to 8938 – can be any number
AT+BTN=”Ezurio Headset”	OK	Specify a random Bluetooth address for connection attempts. The address 008098000001 is safe to use as EZURIO have not issued a Bluetooth device with that address

After setting up the module with these commands and power cycling it, it is ready and waiting for a pairing. Initiate pairing from your mobile phone. On success, you should see the response “PAIR 0 nnnnn” on your terminal emulator. Now the module can be used as a headset.

For example, call your mobile phone – but do not accept the call via the phones keypad.

While the phone is ringing you will see that the module has automatically accepted the Bluetooth connection and there are RING indications being sent on a regular basis. The RING is being sent by the mobile phone and NOT the module.

The headset profile specification states that the command AT+CKPD=200 must be sent from the headset to the phone to answer the audio call. Send AT+CKPD=200 and you will see that your phone answers the call and you should then be able to sustain a conversation via a headset connected to the motherboard.