UARTDNG200 application note

Overview

UARTDNG200 allows you to cut the cables using wireless communication with PC, Pda, Mobile phone, etc...

Based on CSR Chipset its Bluetooth parameters are fully configurable by serial interface.

It's suitable for integration in microprocessor systems without operative system since it does not need drivers to work. It also can be used as simple cable replacement for serial communications.

- Industrial CNC
- Medical devices
- Fitness Machines
- Professional GPS systems
- Telemetry systems
- Sensors
- Your application

Applications

Concerning your needs, UARTDNG200 may be setted up for :

- Pure serial cable replacement (2 UARTDNG200 wireless connected to each other)
- Mixed serial cable replacement (1 UARTDNG200 connected to a Bluetooth PC or a PDA)

The next pages reports some examples of setup.

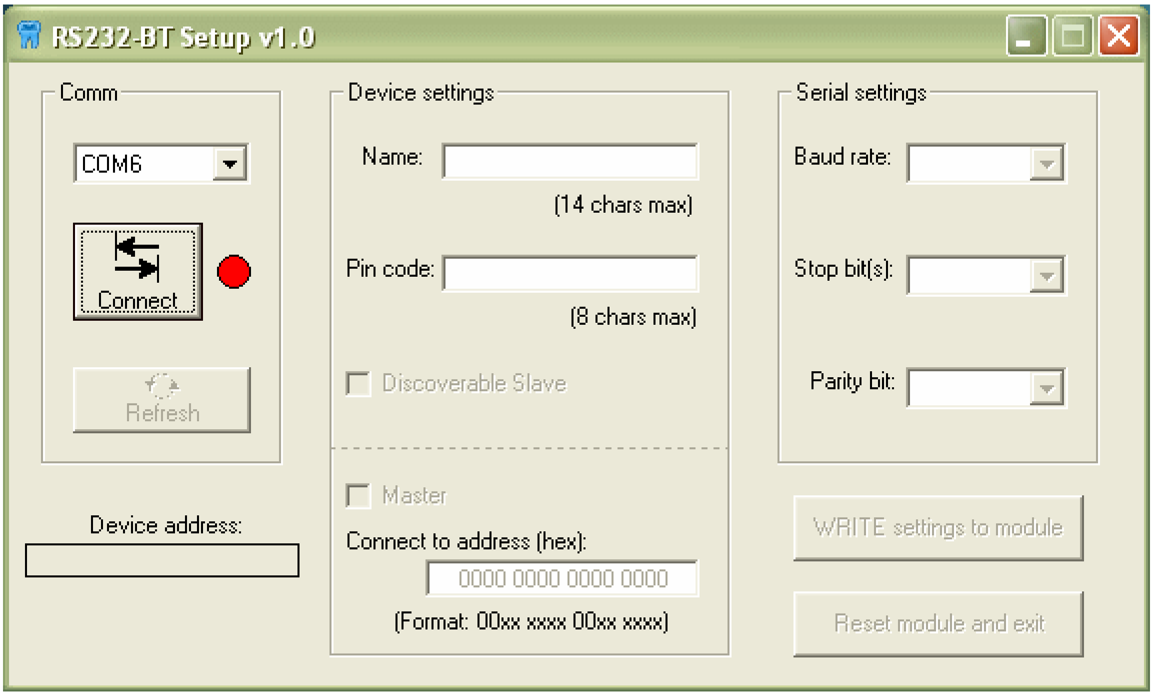
Setup utility

The first thing required is to wire the UARTDNG200 to your PC serial port (RS232_RX, RS232_TX, RS232_RTS[1], RS232_CTS[1] and GND pins). Of course, module will need its own power supply too.

The image below is a snapshot of the setup utility whenever you run it. As you can see, all fields are empty and the status led is red: this means no connection.

You have to select the PC serial port where the module is connected to, then click with your mouse the “Connect” button.

The utility is now waiting for conntection: power on the module and connection with the module will be established. The status led should become green and all fields will be filled with module’s parameters.



Comm: COM6

Connect to address (hex): 0000 0000 0000 0000

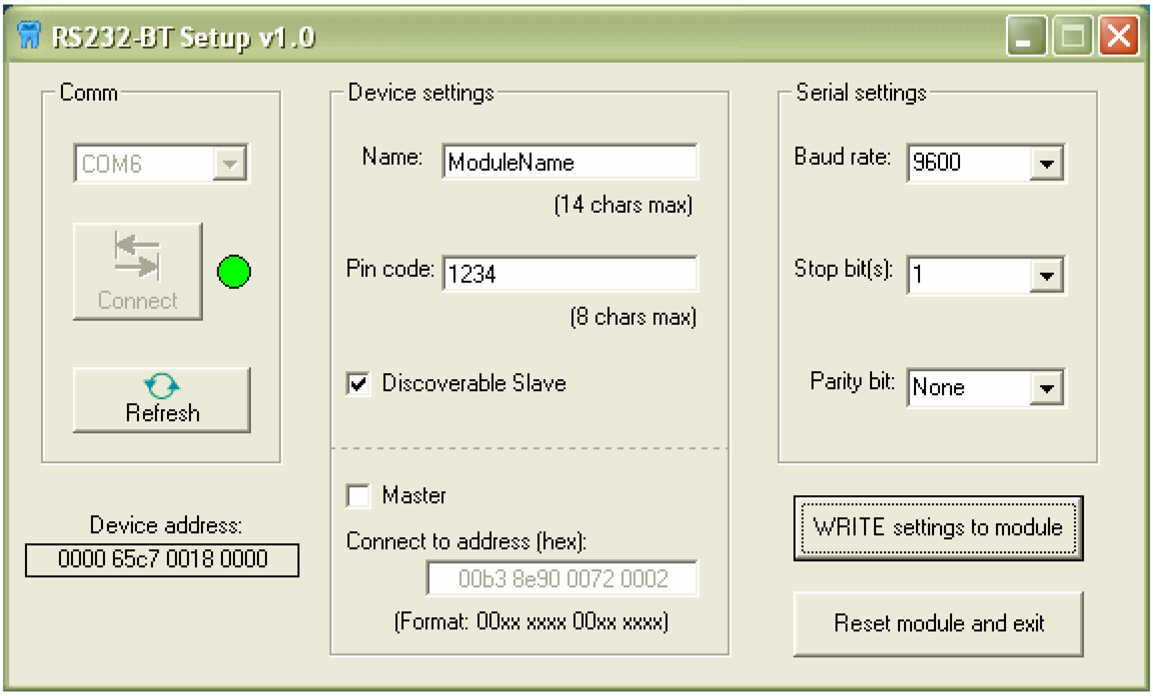
At this point, you are ready to change Bluetooth settings. Next page reports an e-xample.

[1] ***If you don’t need hardware handshake in your project, these pins must be connected together otherwise module will not work properly.***

Bluetooth parameters

The below image reports an example of Bluetooth parameters.

- **Device address**: this is an unique serial number that identify the module anc cannot be changed.
- **Name**: it’s the name that will be shown when the module will be discovered by another bluetooth device.
- **Pin code**: the password code required by module to connect with.
- **Discoverable Slave:** if checked your module (as slave) will be visible (and connectable) to every Bluetooth device; if unchecked only a master that knows slave address can be connected.
- **Master**: if unchecked, module runs as slave serial port profile; if checked, module will become a master and you have to specify the remote device address you want you master device will establish a connection with.
- **Baud rate**: the serial port speed.
- **Stop bit(s)**: the number of stop bits your serial communication uses.
- **Parity bit**: the parity your serial communication uses.



RS232-BT Setup v1.0

Comm
COM6
Connect ●
Refresh

Device address: 0000 65c7 0018 0000

Device settings
Name: ModuleName
(14 chars max)
Pin code: 1234
(8 chars max)
☑ Discoverable Slave

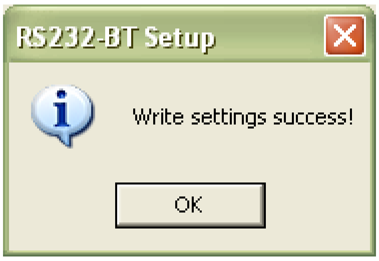
□ Master
Connect to address (hex): 00b3 8e90 0072 0002
(Format: 00xx xxxx 00xx xxxx)

Serial settings
Baud rate: 9600
Stop bit(s): 1
Parity bit: None

WRITE settings to module
Reset module and exit

When done, remember to write the new settings to the module by clicking the “WRITE” buttons. A message window will confirm the success of writing operation.

To make new settings have effect you can click the “Reset” button in the setup utility or switch off and switch on the module.



RS232-BT Setup

Write settings success!

OK

Command line interface

The module’s Bluetooth parameters can be changed by CLI through a serial communication. After power on, the module waits few seconds for a specific command: if received, enters in CLI mode, otherwise runs normally as master or slave device.

These are the available commands:

"**<PING>**"
This command must be sent to the module just after powerd on, within few seconds. It’s used to enter in CLI mode. If repeated, can be useful to verify connection. Module will always reply with “[READY]”.

"**<NAME=abc>**"
Sets device name (max 14 chars).

"**<PKEY=1234>**"
Sets device Pin Code (max 8 chars).

"**<BAUD=nn>**"
Sets the serial speed. The parameter must be a standard value between 1200 and 230400 baud.

"**<SBIT=n>**"
Sets the serial stop bit(s): valid values are 1 and 2.

"**<PRTY=x>**"
Sets the serial parity bit: valid values are N (none), E (even) and O (odd).

"**<SLAVE>**"
Module will run as a slave device.

"**<MASTER>**"
Module will run as a master device. You must specify the slave address too (see next command).

"**<ADDR=hhhh hhhh hhhh hhhh>**"
When the module runs as master, it will attempt a connection to the device with specified address.

"<DISCOVERABLE>" / "<INVISIBLE>"

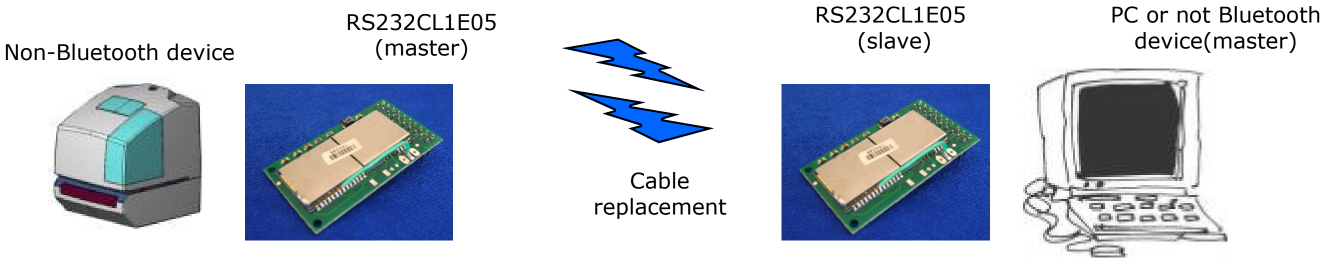
- Slave profile can be visible to every Bluetooth device or invisible; if invisible, only a master that knows slave address can be connected.

"**<RESET>**"
Resets the module (new settings will take effect).

All commands, except <PING> and <RESET> are confirmed with “[OK]” as module reply. To take effect, module needs to be reset (or powered off, then on again) and the new configuration will be used.

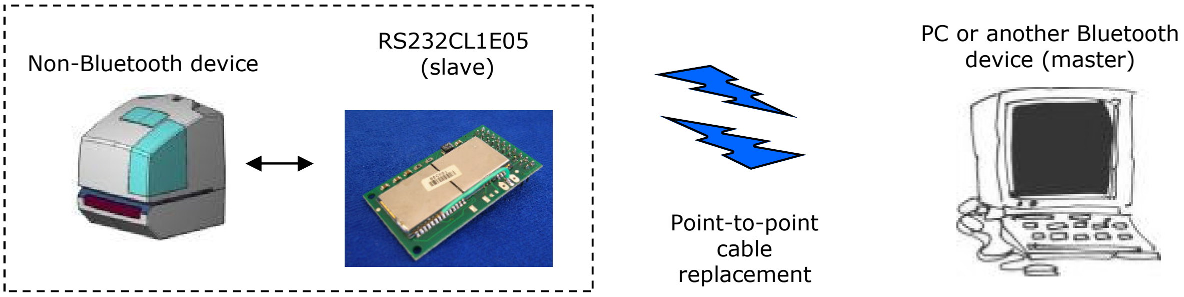
Pure Serial cable replacement solution (master and slave profile):

replace your serial cable with a wireless connection.



Mixed serial cable replacement (slave profile):

wireless connection between a Bluetooth device and a NON-Bluetooth device.



These are typical applications of our BT modules. The issue consists in replacing a physical cable (i.e. RS232 serial cable) with a wireless Bluetooth connection. The communication link can take place without distance problems into the radio visibility area of each Bluetooth far ends.