

BL600 Development Kit

Quick Start Guide

This guide demonstrates how to load a *smart*BASIC Heart Rate Service application (hrs.SB) onto the BL600 development board and view the data on a BT4.0 iPad/iPhone. For this example, the heart rate data originates from a fictional sensor attached to the UART in the format "hr 60".

Requirements

- PC running Windows XP or later
- UWTerminal 6.50 or later
- hrs.heart.rate.SB *smart*BASIC sample application
- USB A to mini B cable
- iPad 3/ iPhone 4S or newer with BT4.0 support
- Internet connection on iOS device (to download the BL600 app)
- DVK_BL600 User Manual
- FTDI Drivers <http://www.ftdichip.com/Drivers/VCP.htm> (for some versions of Windows)

Development Kit Setup

To setup the BL600 development kit (DVK), follow these steps:

1. Configure the BL600 development kit to the following settings:
 - DC/USB power source switch (SW4) set to USB
 - VCC_1V8/VCC_3V3 switch (SW5) set to VCC_3V3
 - CR2033/VCC_3V3/1V8 switch (SW6) set to VCC_3V3/1V8
2. Connect one end of the mini USB cable to CON4 on the DVK board and the other end to your PC.
3. Follow the on screen prompts. Windows may prompt you to install FTDI drivers.

When complete, the DVK board appears in the Windows device manager as a *USB Serial Port*.
4. Note the port number shown in the device manager.
5. Extract UWTerminal and run the program.
6. Configure the COM port with the port number seen in device manager with the following settings (Figure 1):
 - Baudrate – 9600
 - Stop Bits – 1
 - Data Bits – 8
 - Handshaking – None

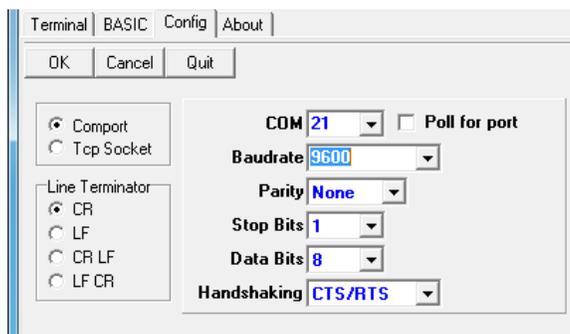


Figure 1: Comms Settings

7. Verify that you can communicate with the board by typing *at* followed by a return. The module responds with *00*.

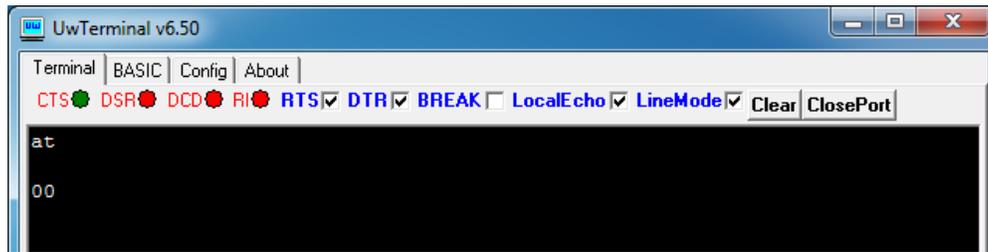


Figure 2: Comms OK

Loading a *smart*BASIC Application

To load a *smart*BASIC application, follow these steps:

1. Ensure that the cross compiler is located in the same folder as UWterminal. Its name is formatted similar to XComp_BL600r2_0B41_1483, where 0B41_1483 indicates a hash key. Every firmware version requires its corresponding cross compiler with a matching hash key.
2. To compile and load a *smart*BASIC application, right-click in the UWterminal main window and select **XCompile + Load**.

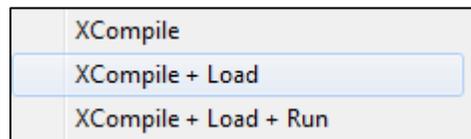


Figure 3: Right-click menu

3. Locate and open the *hrs.heart.rate.SB* application in the supplied *smart*BASIC *_sample_Apps* folder. When the application successfully compiles and loads, the console reads **+++ DONE +++** (Figure 4). If the correct version of cross compiler is not present, an error displays (Figure 5).

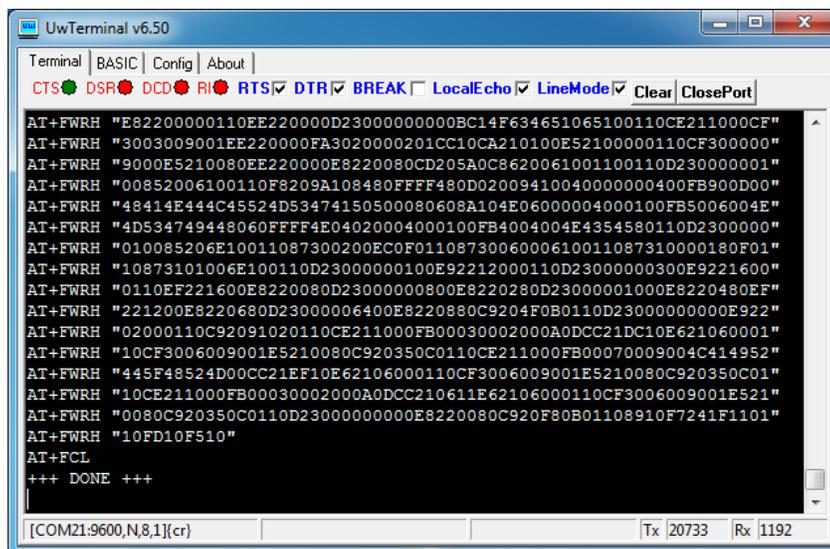


Figure 4: Compiled and Loaded

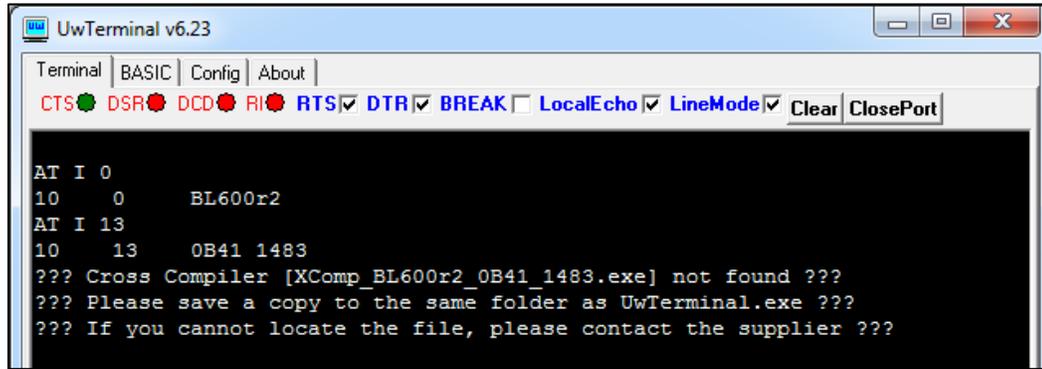


Figure 5: Cross Compiler Error

4. Locate the correct version and place it in the same folder as UWterminal.
5. Confirm that the *hrs* application is loaded by using the command **at+dir**.

Note: The file extension is truncated from files copied onto the BL600 module. Therefore, when *hrs.SB* is copied to the device, its name becomes *hrs*.

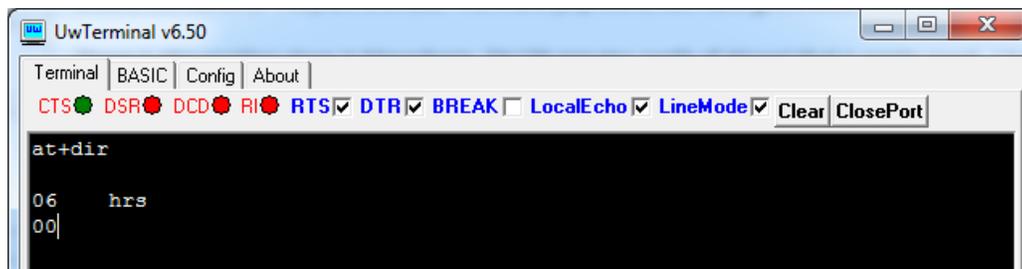


Figure 6: Directory showing "hrs" app loaded

iPad/iPhone Setup

For iPad/iPhone setup, follow these steps:

1. Install the Laird Utility from Apple's App Store and ensure Bluetooth is enabled in the device settings. If you are using an iPad, you may need to click the **iPhone Apps** tab in the App Store.

"BL600" by Laird Technologies - <https://itunes.apple.com/gb/app/bl600/id594855763?mt=8>



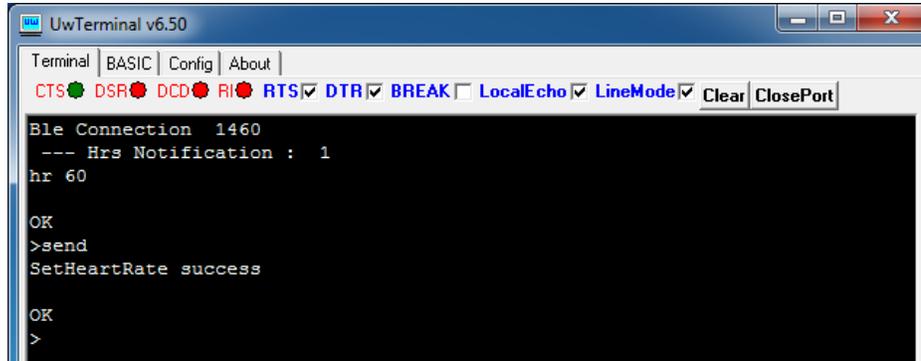
Figure 7: BL600 app installed

2. Once installed, run the BL600 application on your iOS device.

3. Send heart rate data to the iPad using the following case-sensitive command:

```
hr 60<CR>
send<CR>
```

If successful, the module responds `SetHeartRate Success OK >` in UWTerminal (Figure 10).



```
UwTerminal v6.50
Terminal | BASIC | Config | About
CTS DSR DCD RI RTS DTR BREAK LocalEcho LineMode Clear ClosePort
Ble Connection 1460
--- Hrs Notification : 1
hr 60
OK
>send
SetHeartRate success
OK
>
```

Figure 10: Heart rate data sent

The data now appears on the iPad screen (Figure 11):

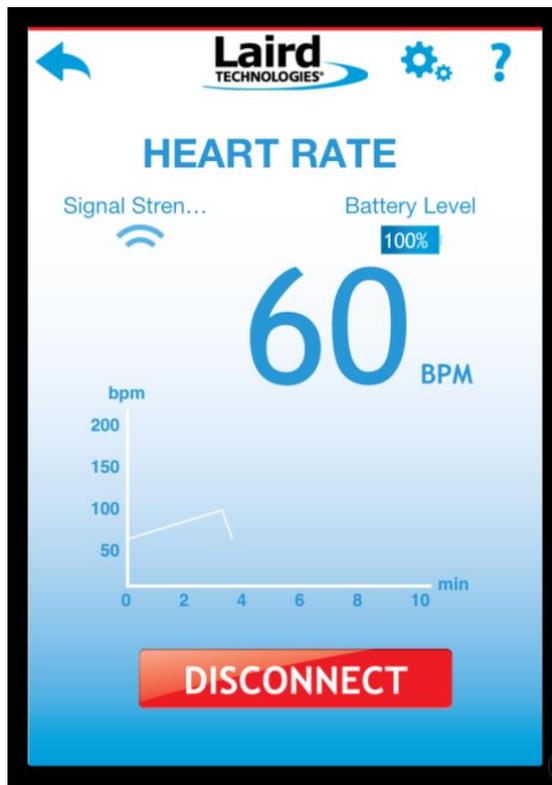


Figure 11: Data received, provided by hrs.sb

To update the iPad with new heart rate values, send the command as in the following example:

```
hr 96<CR>
send<CR>
```

Pressing reset button on the DVK board while hrs.SB is running returns you to the command line where you can reissue commands to the module.