Low Power Modes

BL652 Module

Application Note *v2.0*

1 Introduction

The BL652 module has three power modes:

- **Run** - The normal operation of the module when the application is running and not waiting for events.
- **Standby Doze** -The module is placed in Standby Doze when there are no pending events, and the smartBASIC application reaches the WAITEVENT statement. The module wakes from Standby Doze if any registered event is triggered.
- **Deep Sleep** - Deep Sleep is the lowest power mode, and the module can only exit it through a system reset. It is achieved through the smartBASIC function *SystemStateSet(0)*.

This guide demonstrates how to load a smartBASIC application in order to test the power consumption of the two low power modes: Standby Doze and Deep Sleep.

2 Requirements

- Ezurio DVK-BL652
- Micro USB to USB (Type-A) Cable – provided with devkit
- UwTerminalX v1.06 or later (found at https://github.com/LairdCP/UwTerminalX)
- */p.low.power.deep.sleep.sb* application (found in the BL652 GitHub repository)
- Current measuring digital multimeter (DMM)

3 Development Kit Setup

To set up the DVK-BL652 development kit, complete the following steps:

1. Configure the board to the following settings (shown in Figure 1)
   - DC/USB power source switch (SW4) – in position "USB" (default)
   - VCC_1V8/VCC_3V3 switch (SW5) – in position "VCC_3V3" (default)
   - CR2033/VCC_3V3/1V8 switch (SW6) – in position "VCC_3V3/1V8" (default)
   - Jumper J12 moved to H (pins 1-2 connected)
   - Jumpers J3 and J4 removed
   - Jumper J7 removed
2. To measure the current consumption of the module, connect the current measuring DMM to J7 on the development kit (Figure 1).

Note: After the test is completed, remember to replace the jumper on J7.

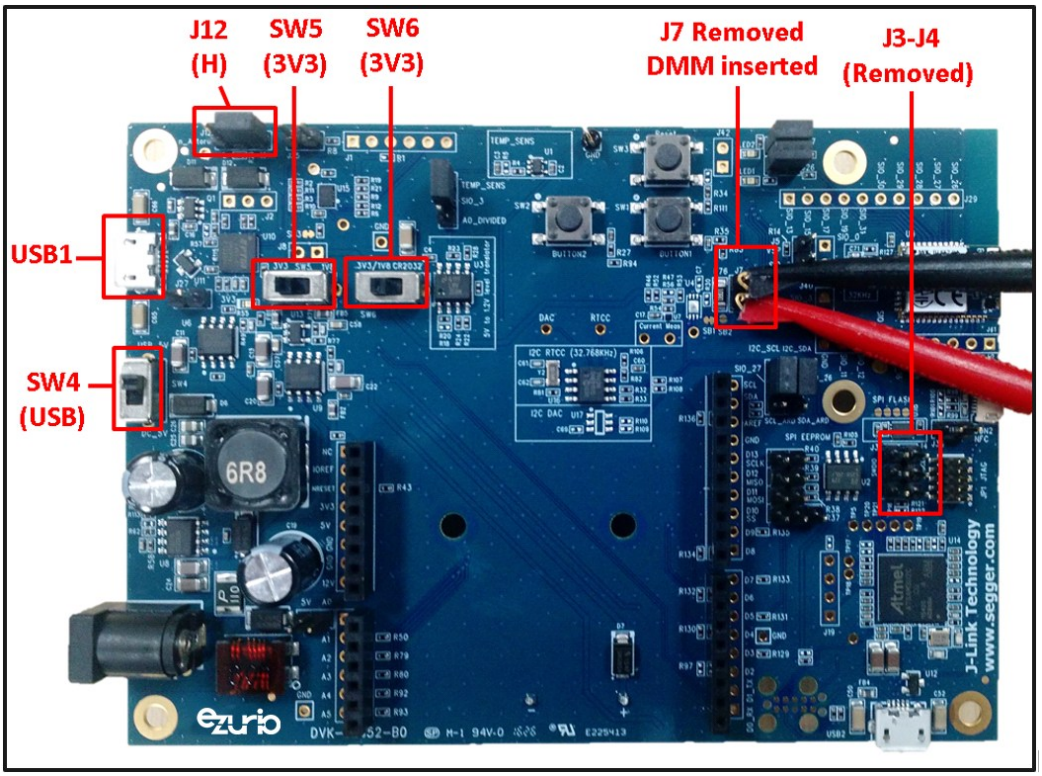

Figure 1: BL652 development board

3. Connect the DVK-BL652 to your PC via the included USB-A to USB Micro cable.
4. Ensure that the current measuring DMM is connected to J7 before the development kit is powered up (or else the board will not be powered).
5. Follow the on screen prompts. Windows may prompt you to install FTDI drivers (found at http://www.ftdichip.com/FTDrivers.htm).
6. Ensure that the Windows Device Manager displays a new virtual COM port for the USB to Serial adapter.
7. Launch UwTerminalX.
8. From the Config tab in Device drop-down menu, select **BL652** to populate the baud, parity, stop bits, data bits, and handshaking settings. If "BL652" is not selectable, set the configuration as shown in Figure 2.
9. In the Port dropdown, select the COM port associated with your DVK-BL652. At the top of the screen, click **OK.**

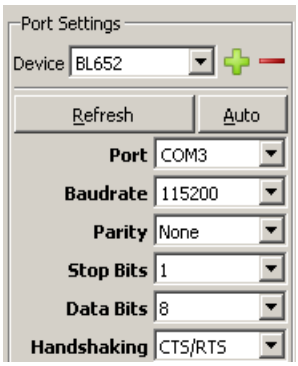

Figure 2: Setting connection parameters

10. To return the BL652 to factory defaults, enter the command *at&f** in the Terminal tab as shown (Figure 3).

Note: If you are using a new development board with the sample application, you may need to remove the autorun jumper on J12 and press the reset button (SW3) to exit out of the sample application, and then issue the *at&f** command to erase the file system and all non-volatile data.

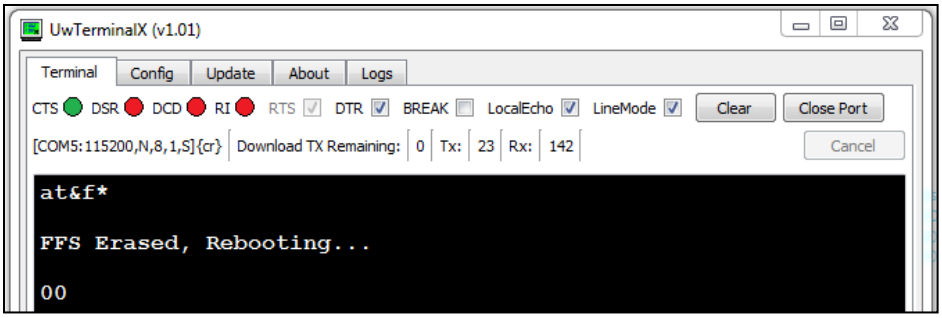

Figure 3: Return BL652 to factory default settings

4 Running Low Power Application

The low power modes application can be downloaded and extracted from the Zip file from GitHub, or saved from the GitHub raw file page: https://github.com/LairdCP/BL652-Applications/blob/master/Applications/lp.low.power.deep.sleep.sb

---

Note: This application note assumes that you are familiar with loading smartBASIC applications onto the BL652 module using UwTerminalX. If not, see the application note "Loading smartBASIC Applications - BL652," available in the Documentation tab of the BL652 product page at ezurio.com.

4.1 Deep Sleep Mode

To measure the deep sleep current, complete the following steps:

1. Since the application is placed into deep sleep mode through *SystemStateSet(0)* smartBASIC function, ensure that the function is not commented out in the file.

```
56  //*************************************************************************
57  // Equivalent to main() in C
58  //*************************************************************************
59
60  PRINT "To get the expected low power consumption please remove jumpers J3 and J4 on your DVK-BL652 board."
61
62  rc=gpiosetfunc(17,2,0)  //Sets LED 0 to digital out low
63  rc=gpiosetfunc(19,2,0)  //Sets LED 1 to digital out low
64  rc=gpiosetfunc(3,2,0)   //Disables the temperature sensor
65  while (uartinfo(5) > 0)
66      //Wait for the UART transmit buffer to empty
67  endwhile
68  uartclose()             //Closes the UART
69  rc=gpiosetfunc(6,2,0)   //TX - set high on default
70  rc=gpiosetfunc(5,2,0)   //RTS
71
72  //Put module into deep sleep
73  rc = SystemStateSet(0)
74
```

Figure 4: Placing module into deep sleep mode

2. Load the Low Power application – use the right-click menu on UwTerminalX window and select **XCompile + Load** and browse to the *lp.low.power.deep.sleep.sb* location on your computer.
3. You can ensure that the application is loaded by issuing *at+dir* as shown in Figure 5:

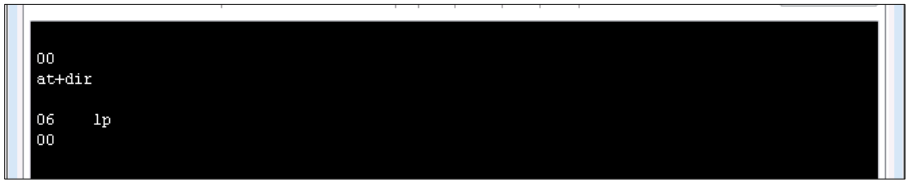

Figure 5: Checking file system

4. Run the low power application by typing *at+run "lp"*.

5. Through the current measuring DMM, current should fall to about 0.4uA (Figure 6).

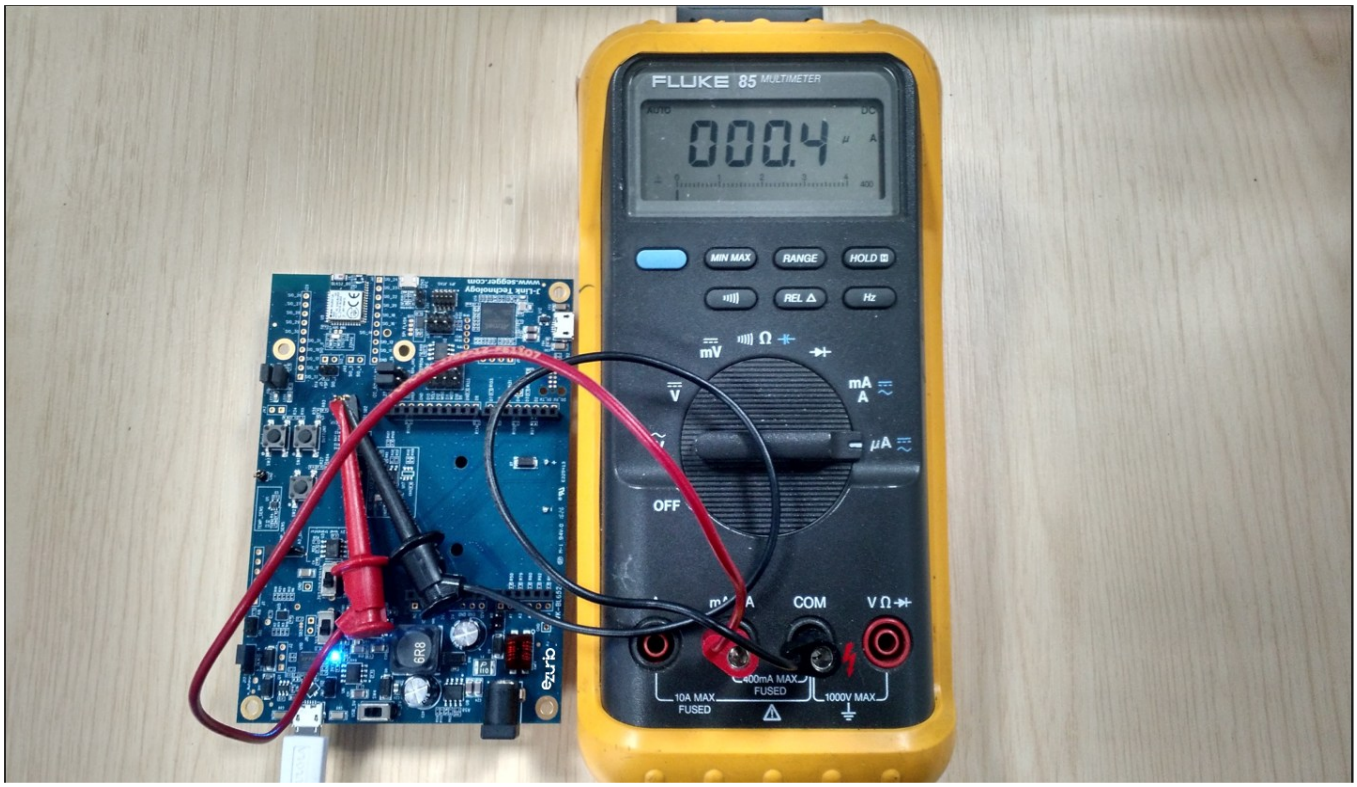

Figure 6: The current falls to about 0.4uA (400nA) when placed in deep sleep mode

Note: The only way to exit deep sleep mode is through a hardware reset. This returns the BL652 into interactive mode or runs an $autorun$ application if one exists in the file system.

4.2 Standby Doze Mode

To measure the standby doze mode current, complete the following steps:

1. The BL652 is placed into standby doze mode when the smartBASIC application reaches WAITEVENT and there are no pending events. To achieve this through the low power mode application, comment out the *SystemStateSet(0)* line in the lp.low.power.deep.sleep.sb file as shown in Figure 7.

```
62  rc=gpiosetfunc(17,2,0)  //Sets LED 0 to digital out low
63  rc=gpiosetfunc(19,2,0)  //Sets LED 1 to digital out low
64  rc=gpiosetfunc(3,2,0)   //Disables the temperature sensor
65  while (uartinfo(5) > 0)
66      //Wait for the UART transmit buffer to empty
67  endwhile
68  uartclose()             //Closes the UART
69  rc=gpiosetfunc(6,2,0)   //TX - set high on default
70  rc=gpiosetfunc(5,2,0)   //RTS
71
72  //Put module into deep sleep
73  //rc = SystemStateSet(0)
74
75  //-----------------------------------------------------------------------------
76  // Wait for a synchronous event.
77  // An application can have multiple <WaitEvent> statements
78  //-----------------------------------------------------------------------------
79  waitevent       //Waits for an event to happen
80
```

Figure 7: Placing module into standby doze mode

2. Reset the BL652 using the reset button (SW3).
3. Erase the flash file system using the interactive command *at&F**.
4. Load the Low Power application – use the right-click menu on UwTerminalX window and select **XCompile + Load** and browse to the *lp.low.power.deep.sleep.sb* location on your computer.
5. Run the low power application by typing *at+run "lp"*.
6. Through the current measuring DMM, the current should fall to about 2.1uA (Figure 8).

Note: Once you're done with the current measurement, place the jumper back on J7 as otherwise the BL652 will not powered (through the USB port) and will be powered parasitically instead.

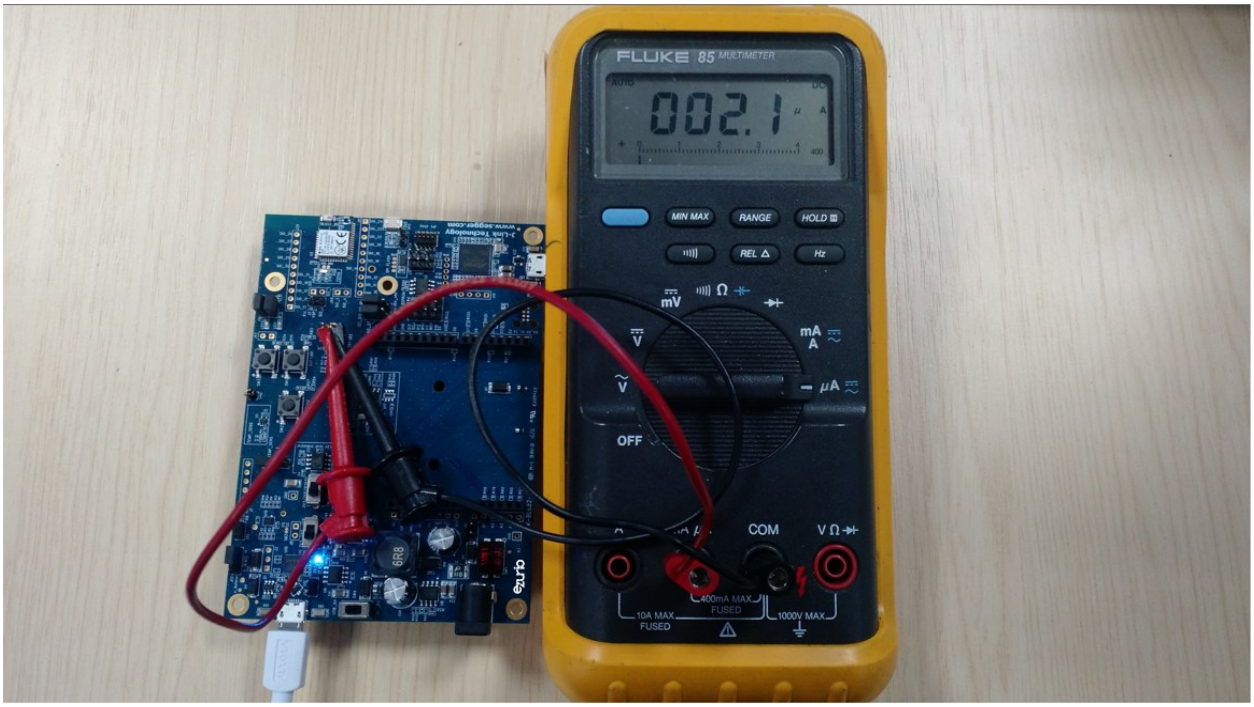

Figure 8: The current falls to about 2.1uA when placed in standby doze mode

5 References

The following documents are also accessible from the Documentation tab in the BL652 product page of ezurio.com:

- Product Brief - BL652
- Datasheet - BL652
- Release Notes - BL652 Firmware v28.6.1.2-R0.
- BL652 Dev Kit User Guide
- User Guide - smartBASIC Core Functionality
- User Guide - BL652 smartBASIC Extensions
- Application Note - What's New in Bluetooth 4.2
- Application Note - Upgrading BL652 Firmware via UART
- Application Note - Getting Started with DVK-BL652 and Nordic SDK Development using Eclipse and GCC
- Application Note - UwTerminalX Quick Reference Guide
- Application Note - Loading smartBASIC Applications - BL652
- Application Note - NFC Manager - BL652 Sample smartBASIC Application
- Application Note - Using Notepad++ and Color Syntax Highlighting - BL652
- Application Note - Using Textpad and Color Syntax Highlighting - BL652

6 Further Information

Further information relating to the BL652 module is available from the BL652 product page at http://www.ezurio.com/products/bl652-ble-module.

7 Revision History

Version	Date	Notes	Approver
1.0	13 Sep 2016	Initial Release	Jonathan Kaye
2.0	20 Mar 2025	Ezurio rebranding	Dave Drogowski

Ezurio's products are subject to standard Terms & Conditions.